School of Information Technology and Engineering
ITE1002 – Web Technologies Laboratory
Assessment-4

Using Node JS, Express JS, Mongo DB

1. Consider a client requests the server to view his profile, which is available in "19BIT0---info.html". Implement it using the Node.js file system module.
2. Develop a small application for the Basketball scoreboard using Node.js event handling. Here, we have a scoreKeeper.on() event-listener listening for an event that would indicate a basket being made. Each scoring attempt is a call of the shoot_a_basket function. If the ball goes in the net (indicated by a true value for the shot), scoreKeeper.emit() is called, which alerts all event-listeners listening for the make_basket event announcement to run their callback function, and passes the value of the basket made. Display the scores of each team.
3. Create a MongoDB "Student" database and do the following operations:
    1. Insert a new student with a name: Dora
    2. Insert a new student with a name: Sinchan and id=2 (integer)
    3. Insert a new student with a name: Angush, the midterm score of 80, and a final score of 100. Scores should be embedded in a sub-document like this: scores:{midterm:0,final:0}.
    4. Finding a document by a single attribute as name as "your name".
    5. Display the list of students who scored between greater than 50 in the midterm.
    6. Search for students that have scored between [50,80] in midterm AND [80,100] in the final exam.
    7. Update the student Sinchan that you created back in exercise 2 to have a midterm score of 50 and a final score of 100 respectively.
    8. Sort according to the final score in descending order and display name and score without objectID.
    9. Delete user Angush that you created back in exercise 3
    10. Delete all users with a midterm score of less than 80.
4. i) Design an application using node.js and MongoDB to perform the following operations in a student collection with a name, age, DOB, and year of admission.
    ii) Insert multiple records into the collection.
    iii) Sort all documents in the student collection

iv) Update the document of a student with name='Kevin' to age=25

v) Limit the result to 4 documents

vi) Query the document based on age>25

5. i) Create a web application with username in HTML and node.js using the express framework to handle different types of HTTP requests namely get, post, put, options, and delete.

ii) Provide different route paths for each of the requests.

iii) Display the different request types with the username in the browser when the application is executed.

6. i) Write an HTML and node.js program for creating and storing session information of the user. The HTML page contains username, password, remember me next time checkbox option and a login button.

ii) Assume, the user name and password are already registered in the node.js server.

iii) Perform the following:

a. If the user enters an invalid user username or password, display an appropriate error message.

b. After successful login, display a welcome message.

c. Allow the user to enter the username and password for 3 times.

If the user enters username and password more than 3 times, display the message "you are blocked".

7. A company database maintains the vehicle information of their employees. It stores the information empid(int), vehicle number(string/int), owner name(string), brand name(string) and year of purchase(int).

a) Create a Collection in MongoDB with the above fields and insert 10 documents at least.

b) Create an HTML form using NodeJS and express for the employee to change the details when he/she changes his/her vehicle by submitting his/her empid and the new vehicle details. The form creation should use CSS for making it interactive and Use ExpressJS at the server-side.

8. The IPL website has a MongoDB database of players. The following information about the players are stored – name, IPL franchise, country, bid_amount

   a) Create an HTML form with appropriate CSS containing text field. Name the text field as **find** and radio button(IPL, country, bid). Name the radio button as **find_details**. On submitting an Express JS in Node server-side code is called that displays information about

   b) The player if the name of the player is entered in **find** and no radio button is checked.

   c) The players of a particular country representing IPL. If the radio button IPL is clicked and country name is entered in **find**

   d) The player name, IPL franchise, and country for the player whose bid amount is greater than or equal or bid amount given in **find**. if the bid radio button is checked and the bid amount is entered in **find**.

   e) Store the data in MongoDB database

Name	IPL Franchise	Country	Bid_Amount
M.S.Dhoni	Rising Pune Super Gaints	India	500000
Raina	Gujarat Lions	India	50000
Bravo	Gujarat Lions	West Indies	200000
Chris Gayle	Royal Challengers Banglore	West Indies	100000
du Plessis	Rising Pune Super Gaints	South Africa	150000
Virat Kohli	Royal Challengers Banglore	india	200000
David Warner	Sunrisers hyderabad	Australia	100000
Sunil Narine	Kolkota Knight Riders	SriLanka	160000