Sr. No.	Title	Domain
1	Write a NodeJS script to take 2 elements 1 & 1000 using file system module & find Kaprekar numbers between them. A Kaprekar number is a number whose square when divided into two parts and such that sum of parts is equal to the original number and none of the parts has value 0.	NodeJS
2	Write a NodeJS script to copy content of one file to another file using asynchronous callback. Copy file content from source.txt to destination.txt. Sequence must be maintained.	NodeJS
3	Develop a user signup form for a website using Express.js and cookies. Implement the following requirements: Create a form with the following fields: oName (input field) oContact Number (input field) oEmail (input field) oAddress (textarea field) oGender (radio buttons: Male, Female, Others) oDOB (date picker) oSubmit button When the user submits the form, store their information (name, contact number, email, Address, gender & DOB) in a cookie named "registered" that expires in 15 seconds. Display a confirmation message to the user after successfully submitting the form & Create a link to display the details stored in the "registered" cookie. When the user clicks to the link, retrieve the information from the cookie and display it on the page also include a link on the details page to Logout. When the user clicks the link, user redirected to home page.	ExpressJS

4	Write express js script to load student form using pug file which contains following fields:
	Roll No(number)
	Name(text)
	Division(text)
	Email(email)
	Subject(radio:FSD-2,COA,PYTHON-2,DM,TOC)
	Once form submitted then data must be displayed on ‘/data’ page using pug file. Means data should be submitted from express application to PUG file.
		ExpressJS

5	Write a program to upload a text file upto 1MB size only using express JS. Perform necessary validation for file format and size.	ExpressJS
6	Write a script to meet following requirements: (1) Create an index.html file & open it on localhost. (2) After clicking submit button, it should jump to "savesession" page. Store username in session. (3) After saving session, redirect to "fetchsession" page & read session value. Put a logout link button here. (4) Jump on "deletesession" page on clicking "logout" link. (5) Destroy the session on this page & redirect to index.html page.	ExpressJS
7	Write a node.js script to load one simple.html file on a NodeJS server & print its contents. HTML file should be opened on localhost & designing should be according to HTML markup.	NodeJS

8	You are tasked with creating a basic Express.js application for a small online store. The application should have the following features: 1) Create a server using Express.js that listens on port 3000. Set up a route to display a welcome message on the homepage ("/") when a user visits it. 2)Create a route to display a list of products ("/products"). You should define an array of product objects with properties like "name," "description," and "price." 3)Create a route handler that renders this list of products in an HTML format. Implement a dynamic route for product details ("/products/:id"). 4)Handle 404 errors by displaying a custom error page when a user tries to access a non-existent route. 5)Your task is to write the Express.js code to achieve the above functionality. 6)Please make sure to include all the necessary dependencies and set up the Express application correctly. Note: You can use any template engine (e.g. Pug) of your choice to render HTML pages, or you can send plain HTML as a response. Ensure that you have installed the required packages and set up the project structure accordingly. Write the Express.js code to implement the features mentioned above.	ExpressJS & NodeJS
9	You are developing a simple Express.js application for a weather forecast service. The application should have the following features: 1) Create an Express.js server that listens on port 3000. 2) Set up a route ("/") that displays a basic welcome message when a user visits the root URL. 3) Create a route ("/weather") that accepts a query parameter "location" and returns the weather forecast for the specified location. You can assume that the weather data is available as an object with properties like "location," "temperature," and "description." For example, if the user accesses "/weather?location=NewYork," the server should respond with the weather information for New York. 4)Handle cases where the user doesn't provide a location query parameter or provides an unknown location by displaying an appropriate error message. 5)Your task is to write the Express.js code to implement these features. Make sure to include all the necessary dependencies and set up the Express application correctly. 6)Note: You can use any template engine (e.g. Pug) of your choice to render HTML responses, or you can send plain JSON responses.	ExpressJS & NodeJS
10	You are building a simple web page that includes a button with the ID "myButton. " You want to display an alert message when the button is clicked. Write the node js code to achieve this using event handling. Instructions: Create an HTML file with a button element that has the ID "myButton." Write the JavaScript code to add an event listener to the button element. When the button is clicked, display an alert message that says, "Button Clicked!"	Node JS & ExpressJS
11	Write a script to Initialize two variables and increment both the variables each time and display the addition of both the variables at interval of 1 second.	NodeJS
12	Write express js script to load student form using pug file which contains following fields Name(text),Email(email),Course(radio : CE, IT, CSE).Once form submitted then data must be displayed on '/student' page using pug file.Means data should be submitted from express application to PUG file.	ExpressJS
13	Write node js script to handle events as asked below.1) Check the radius of circle is negative or not. If negative then display message "Radius must bepositive" else calculate the area of circle. 2) Check side of square is negative or not. If negative then display message "Side must be positive" else calculate the perimeter of square.	NodeJS

14	Write a node.js script to create two listeners for a common event and call their respective callbacks. Print number of events associated with an emitter. Remove one of the listeners & call remaining listeners again. Print number of remaining listeners also.	NodeJS
15	Write a code snippet to configure the multer middleware to store uploaded files in a specific directory called "uploads"	ExpressJS
16	Process a form using post method. Form has fields like username, password, confirm password, gender, submit and reset buttons. After entering all fields, If password and confirm password matches, then form should be processed and all relevant and selected fields’ values should be printed. Otherwise, by printing warning message in red color, it should terminate. No need to write file having form elements.	Express JS
17	Write a program to develop a ToDO list using reactJS.	React JS
18	Write a program to develop a digital clock running continuously using reactJS.	React JS
19	"Create a React component called StockDetail that allows users to input and display stock information. The component should have the following features: Input fields for the following stock details: Name Purchase Price Purchase Quantity Selling Price Selling Quantity When the user clicks the ""Add Stock"" button, the component should: 1. Check if the selling quantity is greater than the purchase quantity it will display an alert message: "Selling quantity cannot be more than purchase quantity." 2. If the selling quantity is equal to or less than the purchase quantity, add a new stock entry into a table below the input fields. Table contains detail of Name,Purchase Price,Purchase Quantity,Selling Price,Selling Quantity and Profit/Loss 3. Display ""Invested"" in the Profit/Loss column if the selling quantity is less than the purchase quantity; otherwise, calculate and display the amount of profit or loss . 4. Style the profit/loss value in green if it's a profit and in red if it's a loss. Write the StockDetail component and any necessary state management code. You can use required React and CSS as per your style."	React JS

20	You are tasked with enhancing the VegetableCategories React component to manage and display vegetable categories and their respective vegetables. Follow the instructions below to complete the task: Entry level category fruits: ['Tomato', 'Cucumber', 'Bell Pepper'], leafyGreens: ['Spinach', 'Kale', 'Lettuce'], rootVegetables: ['Carrot', 'Potato', 'Beetroot'], 1. Default Selection: Modify the component so that when it initially loads, the ""Fruits"" category is selected by default in the dropdown list. [Having random three fruits in that category] 2. Add a New Category: Expand the vegetableData object to include a new category called ""Legumes."" This category should contain the following vegetables: ""Lentils,"" ""Chickpeas,"" and ""Black Beans."" 3. Display New Category: Adjust the component to show the ""Legumes"" category and its associated vegetables when it is selected from the dropdown list. Error Handling: Implement error handling to address two scenarios: If the selected category does not exist in the vegetableData object, display an error message saying ""Category not found."" If the dropdown is set to the default ""Select Category,"" display a message saying ""Please select a category."" Styling: Enhance the component's visual appeal by applying CSS or a CSS-in-JS approach. Your task is to make the necessary modifications and improvements to the	React JS
21	Create a replica of port 27019 to port 27020 in a replica set "rs1". Create collection named “student” having fields name, age, date on primary port. Insert 4 documents. Read documents having age>15 on replica port.	MongoDB
22	Create a form.html file having name, email, date of joining fields and insert data entered by user in collection named “details” in mongoDB. Define a schema having fields like name, email, date of joining. Apply following validations: (1) Name field must remove leading/trailing spaces, minimum and maximum length should be 3 & 12 respectively and it should store values in capital letters only. (2) perform email validation on Email field. (3) date of joining should allow values between 1-1-2010 to 31-12-2022 only(date data type).	MongoDB
23	Consider following citizens collection: [{name: “DD", surname: "abc", age:61}, {name: "LL", surname: "def", age:38}, {name: "KK", surname: "pqr", age:29} {name: "ZZ", surname: "xyz", age:62}] Insert above data in userdata collection under database named “maindata” and write a query to perform below task in node.js. (1) Insert a category field with value “SeniorCitizen” having age greater than 60. (2) Sort the collection by age in descending order and display the youngest person’s name only. (3) Display total number of documents having age between 30 and 60 only. (4) Display only surname field in ascending order of age. (5) Delete the record having age greater than 60.	MongoDB

24	Write a program to build React app to perform the tasks as asked below: 1) Add three buttons “Change Text”, “Change Color”, “Hide/Show”. 2) Add heading “Hello” in red color(initial) and also add “Good Morning” text in h2 tag. 3) By clicking on “Change text” button text should be changed to “Welcome” and vice versa. 4) By clicking on “Change Color” button change color of text to “blue” and vice versa. This color change should be performed while double clicking on the button. 5) Initially button text should be “Hide”. While clicking on it the button text should be changed to “Show” and text “Good Morning” will not be shown.	React JS
25	Basic To-Do List with MongoDB Integration Create a React component for the task management application. This component should have the following functionality: 1. Display a list of tasks. 2. Provide an input field to add new tasks. 3. Implement a button to mark a task as completed or incomplete (toggle the completion status). 4. Display buttons or links to filter tasks by "All," "Completed," and "Incomplete." 5. Insert all these task details into mongoDb table named tasklist inside database named Task	React JS+MongoDB
26	Expense tracking application: Create a React component for the expense tracking application. The component should have the following features: 1. Display a list of expenses, including their names, amounts, categories, and dates. 2. Provide input fields to add new expenses, including name, amount, category, and date. 3. Implement a button to submit and add a new expense. 4. Display dropdowns or buttons to filter expenses by category and date. 5. Handle errors gracefully, such as displaying a message when expenses cannot be loaded or when adding a new expense fails.	React JS
27	Weather Application Create a React component for the weather application. The component should have the following features: •Use pre-defined set of data for weather like :city name,temperature and weather description like cloudy or clear etc. •Allow the user to enter the name of a city into an input field. •Display the current weather conditions specific selected city. •Handle errors gracefully, such as displaying a message when the city name is not found or there is a network issue.	React JS
28	Create a script using frontend technology (ReactJS) to have a form with following fields. --City(Dropdown Box) [Ahmedabad,Rajkot,Surat,Vadodara] --Blood Group (Radio button) [O+,A+,B+,AB+] --Submit button. After clicking on submit button, insert those submitted values of fields in database. Database name - "LJU". Collection name must be "User".	React JS+MongoDB

29	Write a ReactJS program in which a Subject index is given in **tabular** form,on clicking any of the content from the index, the respective page will be displayed with relevant content. Suppose, Subject is FSD2. **Content** JSON NodeJS/ExpressJS React JS Above content Must be in table form If any other link than the specified path is clicked then redirect user to the "No page" link.	React JS
30	Create a script using frontend technology (ReactJS) to have a two fields and a submit button. Fields are Subject Name (Dropdown Box) [FSD2, FCSP2, DS, TOC, COA] (Dropdown box) and Marks(Text Box). After clicking submit button, insert that entered values of fields in database named "StudentData" in collection named "Information".	ReactJS+MongoDB
31	Create a form.html file having Name, Email, City, Date of Exam fields and insert data entered by user in collection named “Exam” in mongoDB. Define a schema having fields like Name, Email, City, Date of Exam. Apply following validations: (1) Name field must remove leading/trailing spaces, maximum length should be 12 and it should store values in capital letters only. (2) Perform email validation on Email field. (3) City must be store in lower letter and values must be any of these (Ahmedabad, Gandhinagar, Vadodara) (4) Date of Exam should allow values between 1-10-2023 to 12-10-2023 only.	MongoDB
32	Consider following employee collection: [{_id: 1,name: "Eric",age: 30,position: "Full Stack Developer",salary: 60000}, {_id: 2,name: "Erica",age: 35,position: "Intern",salary: 8000}, {_id: 3,name: "Erical",age: 40,position: "UX/UI Designer",salary: 56000}, {_id: 4,name: "treric7",age: 37,position: "Team Leader",salary: 85000}, {_id: 5,name: "Eliza",age: 25,position: "Software Developer",salary: 45000}, {_id: 6,name: "Trian",age: 29,position: "Data Scientist",salary: 75000}, {_id: 7,name: "Elizan",age: 25,position: "Full Stack Developer",salary: 42000}] Insert above data in mentioned collection under database named “main” and write a queries to perform below tasks in node.js using Mongoose. (1)Update Documnent with a field "Experience(years)" and value "17" where age is 43 and position "Senior Manager". If no such document is found then insert the document. (2)Display the name and position of the employee with the highest salary (3)Display number of documents where name like “%ric%”. (4)Increase the salary of an employee who has salary less than 45000 by 10%. (5)Display position where name contains only 4 or 5 letters.	MongoDB