

FSD Laboratory 07

Aim: Develop a full stack web application using MERN stack to perform CRUD operations.

Objectives:

1. To develop full-stack web projects using the MERN stack.
2. To learn database connectivity using fetch api.
3. To perform insert, update, delete and search operations on database.

Theory:

1. What is MERN stack?

The **MERN** stack is a popular technology stack for building full-stack web applications. It consists of four key technologies:

- **MongoDB:** A NoSQL database used to store data in JSON-like documents.
- **Express.js:** A back-end web application framework for Node.js that simplifies routing and middleware handling.
- **React:** A front-end JavaScript library developed by Facebook for building user interfaces, especially single-page applications (SPAs).
- **Node.js:** A JavaScript runtime environment that allows developers to run server-side code using JavaScript.

These technologies are all based on JavaScript, enabling full-stack development with a single programming language, both for the front-end and back-end.

2. Use of Fetch API.

The **Fetch API** is a modern JavaScript interface for making network requests, such as retrieving resources from a server. It provides an easy-to-use, promise-based way to make asynchronous HTTP requests (GET, POST, PUT, DELETE) from the client side. Fetch API can handle responses in formats like JSON, and it replaces the older XMLHttpRequest (XHR) for making AJAX calls. Its main use is for interacting with REST APIs and dynamically updating web pages without requiring a full page reload.

FAQ:

1. What makes MERN stack the fastest growing tech stack?

The MERN stack is one of the fastest-growing tech stacks because of several key factors:

- **Full-Stack JavaScript:** MERN uses JavaScript for both client-side and server-side development, making it easier for developers to work with a single language across the entire application.
- **React's Popularity:** React's component-based architecture, virtual DOM, and strong ecosystem make it a powerful and efficient choice for building dynamic user interfaces.
- **Node.js Performance:** Node.js is known for its event-driven, non-blocking I/O, which allows it to handle a large number of simultaneous requests efficiently, making it ideal for high-performance applications.
- **Scalability and Flexibility:** MongoDB's NoSQL structure provides flexibility to handle unstructured data, and its horizontal scalability supports large-scale applications.

- **Vibrant Ecosystem and Community:** The MERN stack has a large community of developers, numerous open-source libraries, and support, making it easier to build and maintain applications.

These features, combined with the stack's alignment with modern web development needs, have contributed to its rapid adoption.

Output: Screenshots of the output to be attached.

Sample Problem Statements:

CRUD Operations using MERN stack:

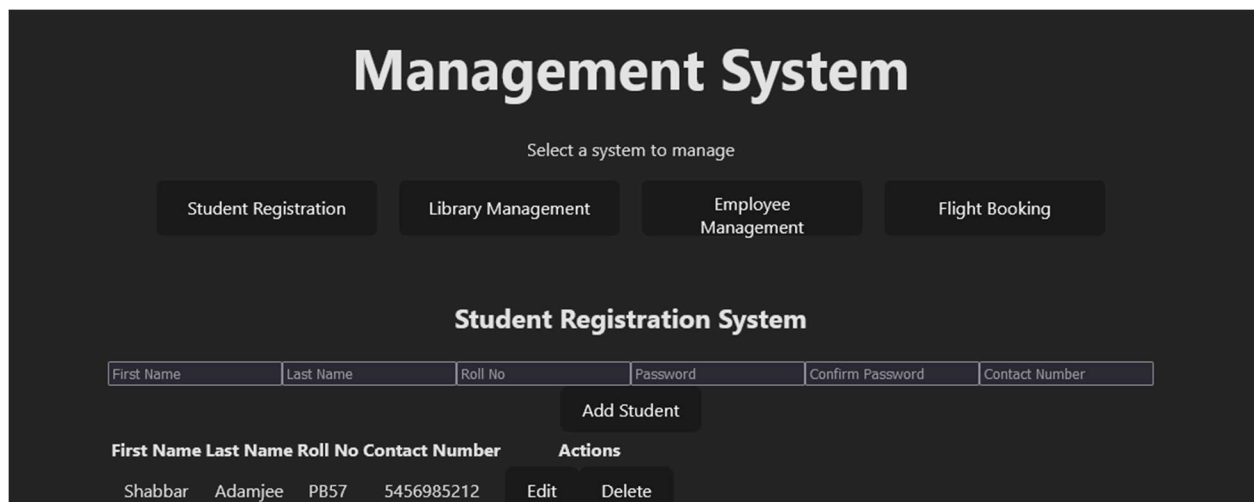
1. Student can create a React form or use existing/ implemented HTML form for Student's Registration System with the fields mentioned: First name, Last name, Roll No/ID, Password, Confirm Password, Contact number and perform following operations

1. Insert student details - First name, Last name, Roll No/ID, Password, Confirm Password, Contact number

2. Delete the Student records based on Roll no/ID

3. Update the Student details based on Roll no/ID - Example students can update their contact details based on searching the record with Roll no.

4. Display the Updated student details or View the Students record in tabular format.



Management System

Select a system to manage

Student Registration Library Management Employee Management Flight Booking

Student Registration System

First Name Last Name Roll No Password Confirm Password Contact Number

Add Student

First Name	Last Name	Roll No	Contact Number	Actions	
Shabbar	Adamjee	PB57	5456985212	Edit	Delete

Management System

Select a system to manage

Student Registration

Library Management

Employee
Management

Flight Booking

Student Registration System

Saurab	Sebait	PB27	5588776633
--------	--------	------	------	------	------------

Add Student

First Name Last Name Roll No Contact Number

Actions

Shabbar	Adamjee	PB57	5456985212	Edit	Delete
---------	---------	------	------------	------	--------

Management System

Select a system to manage

Student Registration

Library Management

Employee
Management

Flight Booking

Student Registration System

First Name	Last Name	Roll No	Password	Confirm Password	Contact Number
------------	-----------	---------	----------	------------------	----------------

Add Student

First Name Last Name Roll No Contact Number

Actions

Shabbar	Adamjee	PB57	5456985212	Edit	Delete
Saurab	Sebait	PB27	5588776633	Edit	Delete

Management System

Select a system to manage

Student Registration
Library Management
Employee Management
Flight Booking

Student Registration System

Saurab
Sebait
PB27
....
....
5588776699

Update Student
Cancel

First Name	Last Name	Roll No	Contact Number	Actions	
Shabbar	Adamjee	PB57	5456985212	Edit	Delete
Saurab	Sebait	PB27	5588776633	Edit	Delete

Management System

Select a system to manage

Student Registration
Library Management
Employee Management
Flight Booking

Student Registration System

First Name
Last Name
Roll No
Password
Confirm Password
Contact Number

Add Student

First Name	Last Name	Roll No	Contact Number	Actions	
Shabbar	Adamjee	PB57	5456985212	Edit	Delete
Saurab	Sebait	PB27	5588776699	Edit	Delete

2. Student can create a React form or use existing/ implemented HTML form for Library Management System with the fields mentioned: Book name, ISBN No, Book title, Author name, Publisher name and perform following operations

1. Insert Book details -Book name, ISBN No, Book title, Author name, Publisher name
2. Delete the Book records based on ISBN No
3. Update the Book details based on ISBN No- Example students can update wrong entered book details based on searching the record with ISBN No.
4. Display the Updated Book details or View the Book Details records in tabular format.

3. Student can create a React form or use existing/ implemented HTML form for Employee Management System with the fields mentioned: Employee name, Employee ID, Department_name, Phone number, Joining Date and perform following operations

1. Insert Employee details -Employee name, Employee ID, Department_name, Phone number, Joining Date
2. Delete the Employee records based on Employee ID
3. Update the Employee details based on Employee ID- Example students can update Employee details based on searching the record with Employee ID.
4. Display the Updated Employee details or View the Employee Details records in tabular format.

4. Student can create a React form or use existing/ implemented HTML form for Flight Booking Management System with the fields mentioned: Passenger name, From, to, date, Departure date, Arrival date, Phone number, Email ID and perform following operations

1. Insert Passenger details -Passenger name, From, to, date, Departure date, Arrival date, Phone number, Email ID
2. Delete the Passenger records based on Phone Number
3. Update the Passenger details based on Phone Number - Example students can update Flight Booking details based on searching the record with Phone Number.
4. Display the Updated Flight Booking details or View the Flight Booking Details records in tabular format.