**FSD Laboratory 07**

**Name : Dev Desai**

**Roll no : 29**

**PRN : 1032211612**

**Panel : F**

**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?

MERN stands for MongoDB, Express, React, Node, after the four key technologies that make up the stack.

* MongoDB — document database
* Express(.js) — Node.js web framework
* React(.js) — a client-side JavaScript framework
* Node(.js) — the premier JavaScript web server

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js is the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

1. Use of Fetch api.

The Fetch API is a feature that allows you to make HTTP requests (such as GET, POST, PUT, or DELETE) to a web server. It's built into modern browsers, so you don't need additional libraries or packages to use it.

Simply put, the Fetch [API](https://blog.hubspot.com/website/application-programming-interface-api?hubs_content=blog.hubspot.com/website/javascript-fetch-api&hubs_content-cta=API) makes it easy to get information from a website and do something with that data in your browser (or whatever environment you're using). For example, you can use the Fetch API to request an [HTML](https://blog.hubspot.com/website/html?hubs_content=blog.hubspot.com/website/javascript-fetch-api&hubs_content-cta=HTML) document from a website and then parse it to get certain elements out.

**FAQ:**

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

JavaScript Everywhere: MERN stack allows developers to use JavaScript across the entire application stack, from server to client. This unified language streamlines development and promotes code reuse.

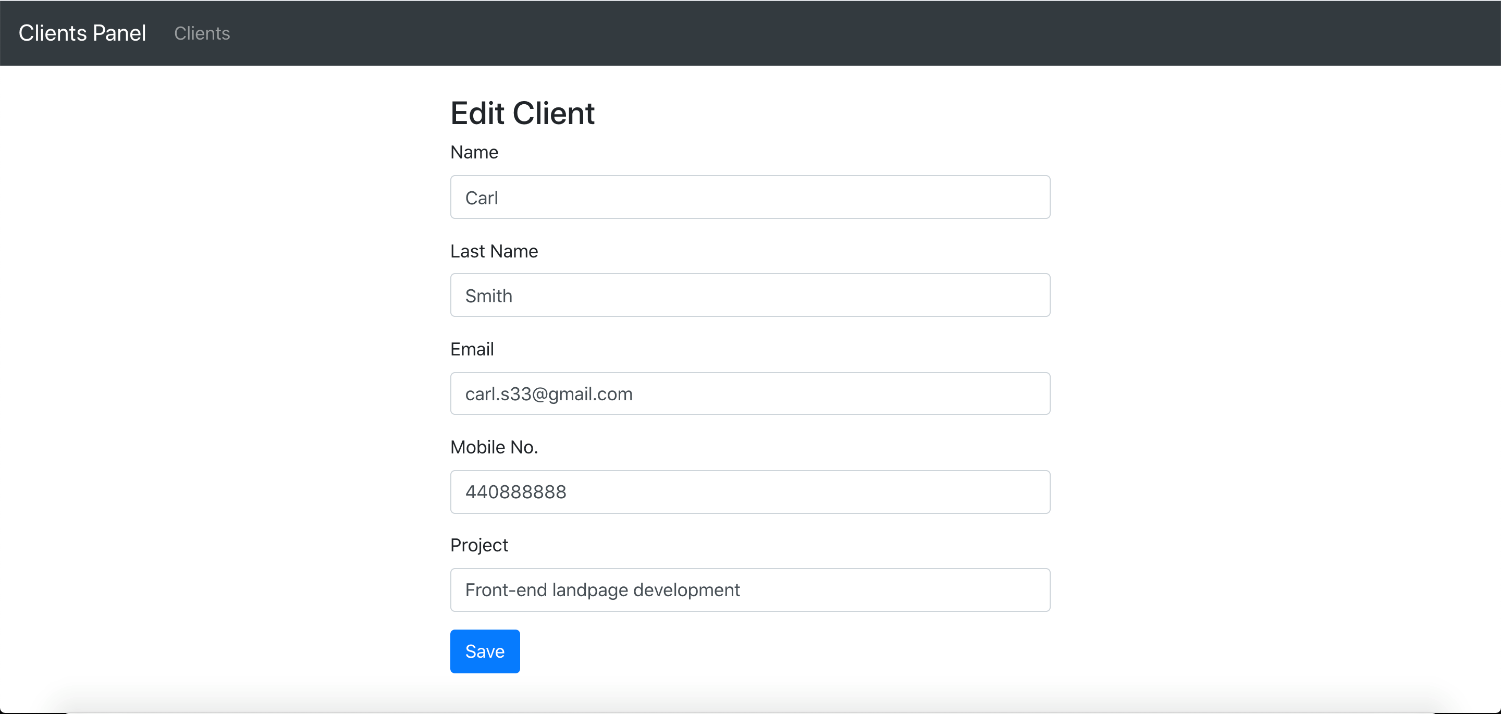
Node.js's Non-blocking I/O: Node.js, a key component of MERN, uses non-blocking I/O, enabling it to handle multiple concurrent requests efficiently. This can result in faster data processing and better performance, especially in applications requiring real-time updates.

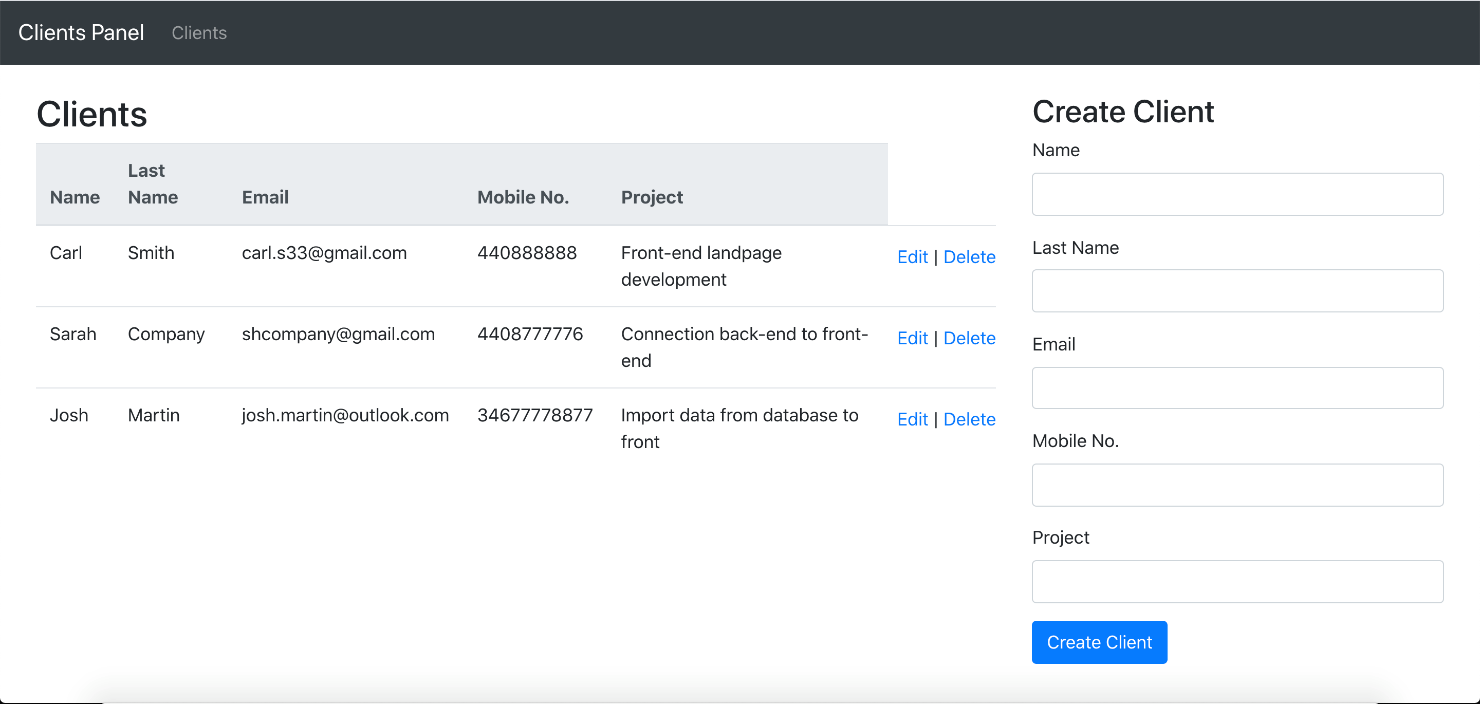
React's Virtual DOM: React, a powerful front-end library in the MERN stack, uses a virtual DOM that enhances performance by minimizing actual DOM manipulation, leading to faster rendering and updates in web applications.

Scalability: MERN's components are known for their scalability. MongoDB, as a NoSQL database, is easily scalable, while Node.js's event-driven architecture and React's component-based structure facilitate the scalability of both the server and client sides.

Community Support: The MERN stack has a large and active developer community that contributes to its ecosystem by creating numerous libraries, modules, and resources. This support network can accelerate development and troubleshooting.

**Output: Screenshots of the output to be attached.**





Help Link:

https://www.mongodb.com/languages/mern-stack-tutorial