

REPORT OF LAB WORK ON BACKEND

**Submitted in partial fulfillment of the requirements for the award
of degree of**

BACHELOR OF ENGINEERING

In

COMPUTERSCIENCE AND ENGINEERING

Submitted by: Aditya Sehra

Roll Number: 2110990099 (G8)

Mentor Name: Dr. Arzoo Miglani



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING CHITKARA UNIVERSITY

Blog Application Report

Introduction

- The Blog Application is a simple demonstration of how to create a totally functioning web application.
- The Admin can add blogs, edit the blogs, and delete the blogs. The application is built on the Node.js back-end and a combination of HTML, CSS, and JavaScript for the front-end.

Backend Development

- The backend for the application was created using Node.js with the help of Express.js. Its main purpose is to manage data and execute the requests coming from the users.
- Setting Up the Backend started with the setup of Node.js and the building structure of the project. On Express, routes were setup toward handling tasks.

Task Management Routes

- A GET route was added to fetch all blogs.
- A POST route was added to allow users to add new blogs.
- A DELETE route was added to allow admin to delete posts by their unique task ID.

Data Base

Database Overview

Database Type: NoSQL, document-oriented database.

Technology: MongoDB, known for its flexibility and scalability.

Database Design

Data Model: Document-based model using JSON-like BSON format.

Collections: Grouping of documents (similar to tables in relational databases).

Documents: Basic unit of data (similar to rows in relational databases), which can include nested fields and arrays.

Key Features

Schema Flexibility: Unlike relational databases, MongoDB allows dynamic schema design. This means you can store different documents in the same collection.

Scalability: Horizontal scaling through sharding, which distributes data across multiple servers.

Running the Server

- I coded to begin the Express server on a specific port so it can respond to requests from the front-end.

Front-end Development

- The front-end of this application was meant to offer user-friendly access to the blogs. This bit of the project was therefore developed using HTML for structure, CSS for styling, and JavaScript for dynamic interactions.

HTML Structure

- A text input box where the users could type in their search.

- The button to add the blogs to the list in the admin panel
 - An admin dashboard with the display of the blogs in a plain list
- Format

Styling with CSS

- Created a centered layout with a card-like white container.
- and made the effects on Hover for better user interaction.

Dynamic Behavior with JavaScript

- It used JavaScript to link the front to the back.
- Each time a admin inputs a blog, it sends the data back to the back-end using an HTTP request; it automatically updates the page to reflect the newly added blog.
- Also included is a delete button on each blog, which enables admin to remove tasks-thereby resulting in sending a request to the back-end which automatically updates the blog list.

Integration of Back-end and Front- end

- There are two parts to an application: a backend and a front-end.
- The integration of these parts makes the application functional overall. Here's how they were connected:

- API Requests
- The front-end sends data such as a new blog to the backend using HTTP requests (with the use of the fetch API).

Conclusion

- Such an app like "Eureka - Blogs" is a simple yet powerful project, which allows developers to demonstrate the basics of web development.
- Going through its creation, I could learn how to make a backend that manages data and requests and to devise a front-end that would work neatly in conjunction with it.
- This is the kind of application that highlights the importance of integrating both parts together for a more user-friendly experience.
- It's a great beginning in understanding how modern web applications work.
- This app is fully functional; add a user authentication system or sort tasks for further strength.