**DESIGN AND IMPLEMENT A MONGODB DATABASE**

**A NAAN MUDHALVAN REPORT**

***SUBMITTED BY***

**ARUNKUMAR K 912421106001**

**ATCHAYA V 912421106002**

**BACHELOR OF ENGINEERING IN**

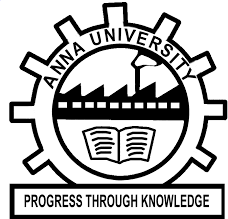
**ELECTRONICS AND COMMUNICATION ENGINEERING**



**SHANMUGANATHAN ENGINEERING COLLEGE ARASAMPATTI, PUDUKKOTTAI – 622 507**

**YEAR & SEMESTER : IV & VII SUBJECT CODE :NM1050**

**COURSE NAME :SaaS (SOFTWARE AS A SERVICE)**



**ANNA UNIVERSITY::CHENNAI 600 025**

**NOV/DEC 2024**

**BONAFIDE CERTIFICATE**

Certified that this Naan Mudhalvan report of **“DESIGN AND IMPLEMENT A MONGODB DATABASE”** is the bonafide work of **“ARUNKUMAR K (912421106001)” & “ATCHAYA V (912421106002)”**who carried out the project work under my guidance.

## SIGNATURE SIGNATURE

**Mrs. D. LATHA M.E., Dr. A.MUTHU MANICKAM M.E., Ph.D., NAAN MUDHALVAN COORDINATOR HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR, ASSISTANT PROFESSOR,**

Department of Electronics & Department of Electronics &

Communication Engineering, Communication Engineering, Shanmuganathan Engineering College, Shanmuganathan Engineering College, Arasampatti – 622 507 Arasampatti – 622 507

Submitted for the Internship viva-voice on

INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGMENT**

At this pleasing moment having successfully completed our internship report, we wish to convey our sincere thanks to our beloved chairperson **Mrs. PICHAPPA VALLIAMMAL**, correspondent **Dr. P. MANIKANDAN B.E,** director(Academic) Shri **M.SHANMUGANATHAN**, director(Administration) Shri **M. PICHAPPA** and honourable secretary **Mr. M. VISWANATHAN** for their extensive support.

I thankful to our principal **Dr. KL. MUTHURAMU M.E(W.R)., M.E(S.E)., Ph.D., FIE., M.I.S.T.E**., Shanmuganathan engineering college, for providing the opportunity to conduct our project.

I extend our gratitude to **Dr. A.MUTHU MANICKAM M.E., Ph.D**., the head of the department and our internship co-ordinator of Electronics and communication engineering for providing a valuable suggestion and supports given through the study.

Gratitude never fails. We are grateful to our dynamic and effective internal guide **Asst. Prof. Mrs. D. LATHA, AP/ECE, M.E.,** for his valuable innovative suggestion, constructive interactions, constant encouragement and valuable helps that have been provided us throughout the project.

I also express my heartfelt thanks to all other staff members of Electronics communication engineering Department for their support. Above all, we thank our parents, for affording us the valuable education till now.

# ABSTRACT

Project focuses on the design and implementation of a MongoDB database to manage and store data efficiently for a hypothetical application. MongoDB, a NoSQL database, offers a flexible, schema-less structure that allows for dynamic data storage and scalability, making it ideal for modern applications. The project involves designing a data model that includes collections for users, activities, and preferences, and implementing the database to support essential operations such as Create, Read, Update, and Delete (CRUD). Key features of the database include indexing for optimized query performance, the use of relationships between collections through ObjectIds, and the scalability to handle large datasets. The implementation is carried out using MongoDB's native driver, and testing is performed to ensure data integrity, performance, and reliability. This project highlights MongoDB's advantages in handling complex, evolving datasets and demonstrates its use in building scalable, high-performance applications. Future work involves enhancing security, backup strategies, and cloud deployment for further optimization.

# TABLE OF CONTENT

## CHAPTER NO TITLE PAGE NO

1. **INTRODUCTION OF MONGODB**
   1. INTRODUCTION 1
   2. KEY FEATURES OF MONGODB 1
   3. USE CAUSE OF MONGODB 2

## TOOLS USED TO DEVELOP IMPLEMENT MONGODB

* 1. TOOLS TO DEVELOP IMPLEMENT

MONGODB 4

* 1. FRONT-END DEVELOPMENT 4
  2. BACK-END DEVELOPMENT 6
  3. DATABASE 6

## PROJECT IMPLEMENTATION

* 1. CODES FOR PROJECT IMPLEMENTATION 7
  2. HTML FOR REGISTRATION PAGE 7
  3. NODE.JS CODE FOR SERVER SIDE 8

## CONCLUSION

* 1. CONCLUSION 10

# CHAPTER 1

**INTRODUCTION OF MONGODB**

**MONGODB:**

MongoDB is a popular, open-source, NoSQL (Not Only SQL) database management system designed for storing and managing large volumes of data. It is document-oriented, meaning that it stores data in flexible, JSON-like format called \*BSON\* (Binary JSON). Unlike traditional relational databases, which store data in tables with rows and columns, MongoDB uses collections and documents, which allows for more flexible, scalable,and high-performance data storage.

**Key Features of MongoDB:**

**1. Document-Oriented Storage:**

Data is stored in collections as documents (in BSON format). A document is similar to a JSON object, but with additional data types like ObjectId, Date, etc.

**2. Scalability:**

MongoDB provides built-in horizontal scalability. It allows data to be distributed across multiple servers through sharding (splitting data across multiple machines), enabling it to handle large datasets effectively.

**3. Flexible Schema:**

Unlike traditional relational databases, MongoDB does not require a predefined schema. Documents within the same collection can have different fields and data types, offering flexibility to evolve your data model without downtime.

**4. Indexing:**

MongoDB supports a variety of indexing strategies, allowing for efficient querying and retrieval of data. By default, it creates an index on the \_id field, but additional indexes can be created to optimize specific query patterns.

**5. Aggregation:**

MongoDB includes a powerful aggregation framework for performing operations like filtering, sorting, grouping, and joining data. This framework provides a pipeline for transforming data, making it easy to perform complex data manipulations.

**6. Replication:**

MongoDB supports replication, where data is replicated across multiple servers in a replica set, ensuring high availability and fault tolerance.

**7. Ad-hoc Queries:**

MongoDB allows complex queries to be executed dynamically without needing predefined SQL-like schema. You can query documents using a variety of operators for conditions, ranges, and other criteria.

**8. Real-Time Performance:**

Due to its flexible design and indexing strategies, MongoDB is optimized for fast read and write operations, making it a strong choice for applications requiring real-time performance.shopping more convenient than ever before.

**Use Cases for MongoDB:**

**1.Content Management System (CMS):**

* MongoDB is well-suited for content-heavy applications that need to store and manage unstructured or semi-structured data. The flexible schema allows for the storage of various types of media (e.g., text, images, videos) and metadata, which is useful for CMSs.
* Example: Websites, blogs, and media portals that handle a variety of content formats.

**2.Real-Time Analytics:**

* MongoDB can handle large volumes of real-time data efficiently, making it ideal for use cases like real-time analytics, dashboards, and monitoring systems. Its ability to scale horizontally allows it to manage high write throughput and read queries simultaneously.
* Example: Real-time business intelligence applications, monitoring tools, and logs analytics platforms.

**3.Big Data Applications:**

* MongoDB’s scalability and support for large datasets make it an excellent choice for big data applications. It can handle vast amounts of data, and with its sharding capabilities, it can distribute the data across multiple machines to ensure performance and reliability.
* Example: Data lakes, big data processing systems, and data warehouses.

**4.Internet of Things (IOT):**

* MongoDB is often used in IoT applications to store time-series data, sensor readings, and device logs. Its flexibility and horizontal scalability allow it to handle massive volumes of data generated by devices and sensors.
* Example: Smart home systems, industrial IoT platforms, and connected devices.

**5.Mobile And Web Applications:**

 MongoDB is commonly used in mobile and web applications due to its ease of integration and support for storing diverse data types (like JSON documents). It is particularly suited for applications that need to scale quickly or handle user-generated content, including social media apps and e-commerce platforms.

 Example: Online marketplaces, social networks, and gaming platforms.

**6.Gaming:**

* MongoDB’s scalability and flexibility make it a good fit for online multiplayer games, where there is a need to store player profiles, game states, leaderboards, and other dynamic data.
* Example: Multiplayer games, online gaming platforms, and in-game virtual economies.

# CHAPTER 2

# TOOLS USED TO DEVELOP IMPLEMENT MONGODB

# TOOLS TO DEVELOP IMPLEMENT MONGODB

## When developing and implementing MongoDB in applications, there are several tools available to facilitate database management, development, data migration, and integration with other systems. Here are some of the key tools you can use for working with MongoDB in the development and implementation process

## FRONT-END:

## HTML

## CSS

## BACK-END:

## NODE .JS

## DATABASE:

## MONGODB

## FRONT-END DEVELOPMENT:

## The ****front-end**** of a web application is the part that users interact with directly. It includes everything that users experience visually and interactively: the layout, design, structure, and behavior of the website or app. In web development, the front-end is typically composed of HTML, CSS, and JavaScript, along with various tools and frameworks that help to streamline development and enhance functionality.

## HTML:

HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It provides the framework that organizes text, images, links, and multimedia content into a structured webpage. HTML uses tags enclosed in angle brackets (< >) to denote different elements, with pairs of tags (an opening and a closing tag) marking the beginning and end of content sections.

## Html icon and Html logo - Download in SVG, PNG, ICO, ICNS

## CSS:

**CSS** (Cascading Style Sheets) is a stylesheet language used to control the layout and presentation of HTML elements on a webpage. It separates the content (HTML) from the design (CSS), allowing for better organization, flexibility, and maintenance of the website's visual presentation.



## USES OF CSS:

1. **Separation of Content and Style:** CSS allows you to define the presentation (fonts, colors, margins, etc.) separately from the HTML content, making the code cleaner and easier to maintain.
2. **Consistency:** With CSS, you can apply the same style across multiple pages, ensuring consistency across your website.
3. **Responsive Design:** CSS enables you to create responsive layouts that adapt to various screen sizes (mobile, tablet, desktop).
4. **Customization:** CSS offers a wide range of styling options to customize the look and feel of your website, including fonts, colors, animations, transitions, and positioning.

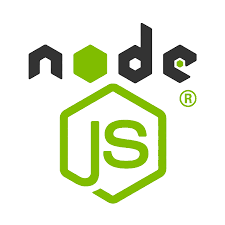
## Types of CSS:

* 1. **Inline CSS**: Applied directly within an HTML element using the style attribute.
  2. **Internal CSS**: Defined within a <style> tag in the <head> section of an HTML document.
  3. **External CSS**: Linked to an external CSS file using the <link> tag. This is the most common method, especially for larger websites.

## BACK-END DEVELOPMENT:

**NODE.JS:**

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to use JavaScript to write server-side code, enabling full-stack JavaScript development. This means you can build backend services, APIs, command-line tools, and more using JavaScript. Node.js is popular for building fast, scalable, and asynchronous applications.



### **Key Features of Node.js**

1. **Event-Driven, Non-Blocking I/O**:
   * Node.js uses an asynchronous, non-blocking I/O model, making it efficient and well-suited for data-intensive applications. This model allows Node.js to handle many requests concurrently by processing requests in a non-blocking manner.
2. **Single-Threaded**:
   * Node.js is single-threaded but highly scalable because it uses an event-driven model to handle concurrent operations.
3. **NPM (Node Package Manager)**:
   * Node.js has a vast ecosystem of packages available through NPM, the world's largest software registry, which allows developers to easily incorporate pre-built modules.

## DATABASE:

In this Project ,here I am using MONGODB with the help of MONGODB COMPASS

## MONGO DB:

MongoDB is a NoSQL, document-oriented database that stores data in a flexible, JSON-like format called BSON (Binary JSON). It’s designed to handle large volumes of unstructured or semi-structured data and is especially popular for applications that require fast, scalable data storage and retrieval, such as web and mobile applications..



# CHAPTER 3

**PROJECT IMPLEMENTATION**

## CODES FOR PROJECT IMPLEMENTATION:

## HTML FOR REGISTER PAGE:

<div class="form-group">

<label for="username">NAME</label>

<input type="text" id="username" name="username" required>

</div>

<div class="form-group">

<label for="email">Email:</label>

<input type="email" id="email" name="email" required>

</div>

<div class="form-group">

<label for="regno">REG NO</label>

<input type="text" id="regno" name="regno" required>

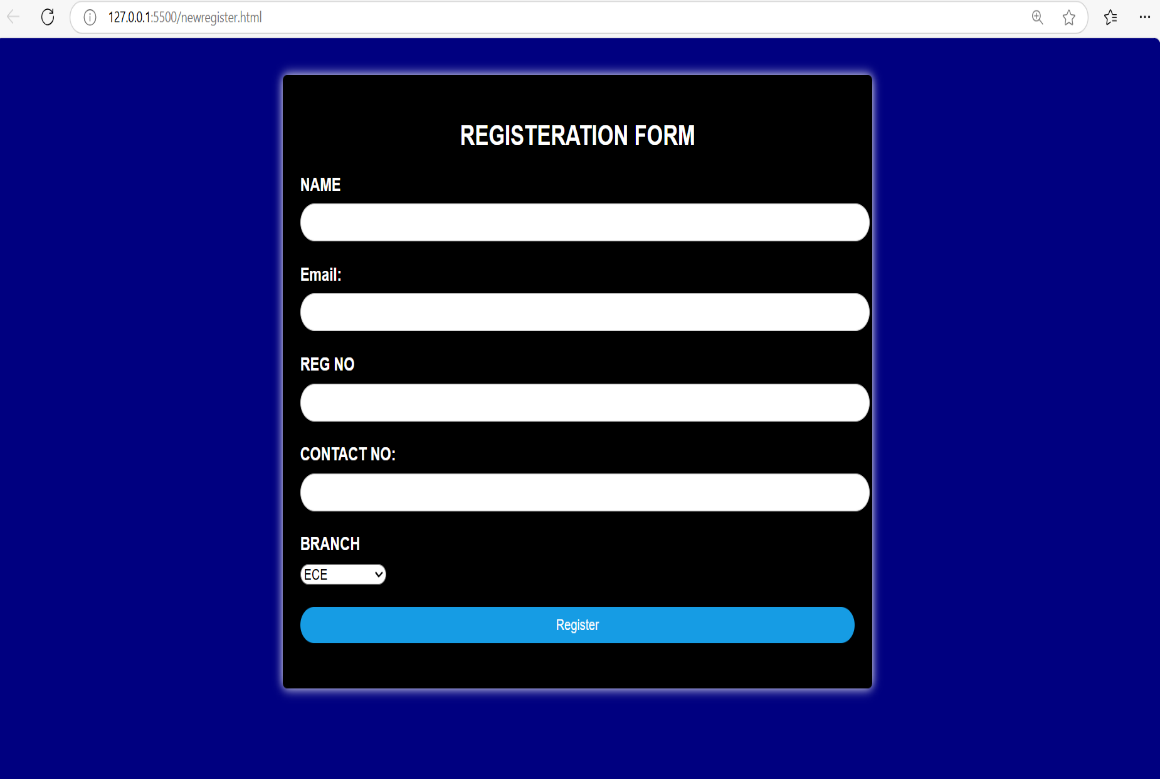
</div>

<div class="form-group">

<label for="contactno">CONTACT NO:</label>

<input type="text" id="contactno" name="contactno" required>

</div>



## NODE.JS CODE FOR SERVER SIDE:

mongoose.connect('mongodb://127.0.0.1:27017/students')

const db = mongoose.Connection

mongoose.connection.once('open',()=>{

console.log("Mongodb connection successful")

})

app.post('/post',async (req,res)=>{

const {username,email,regno,contactno,branch} = req.body

const user = new users({

username,

email,

regno,

contactno,

branch

})

await user.save()

console.log(user)

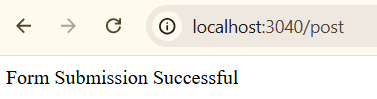
res.send("Form Submission Successful")

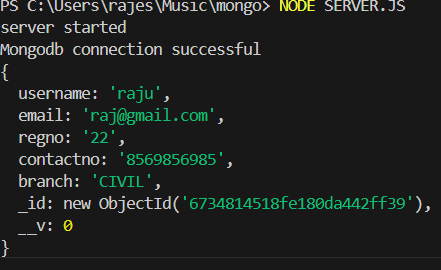
})

app.listen(port,()=>{

console.log("server started")

})





**OUTPUT:**

The Output that we get in the project implementation is the figures ,codes and the HTML

,CSS and relevant Front-end and the Back-end that should be obtain from Visual Studio code.

# CHAPTER 4 CONCLUSION

## CONCLUSION:

Implementing MongoDB can be a highly effective choice for applications that require flexibility, scalability, and rapid development. Its schema-less, document-oriented model makes it particularly well-suited for handling unstructured or evolving data, as it allows for easy modifications and additions without downtime. With powerful features like horizontal scaling (sharding), high availability through replica sets, and a versatile aggregation framework, MongoDB supports complex data operations, high-volume data storage, and real-time analytics.

Additionally, MongoDB integrates seamlessly with modern development stacks and frameworks, especially in JavaScript-based environments like Node.js. This compatibility allows for efficient development cycles and full-stack JavaScript applications, making it popular for web, mobile, IoT, and big data applications. For developers seeking performance, flexibility, and ease of use in a database, MongoDB presents a highly adaptable and scalable solution, enabling businesses to grow while maintaining a simple, developer-friendly database infrastructure.

In summary, MongoDB’s NoSQL design, adaptability, and scalability make it an ideal choice for applications requiring fast data processing, dynamic data modeling, and a high-performance database. It is an excellent option for both startups and established enterprises aiming to build applications that can easily handle evolving requirements and large-scale growth.