**Summer Training Report/ Synopsis/ Minor Project**

**On**

**Student Login and Grievance System**

**In MongoDB**

**A Project Report/Synopsis submitted in partial fulfilment of**

**the requirements for the award of**

**Bachelor of Engineering**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**Taruna Saini**

**(Roll no:CO19366)**

**Under the supervision of**

**(Dr. Dheerendra Singh – Professor in CSE, CCET)**



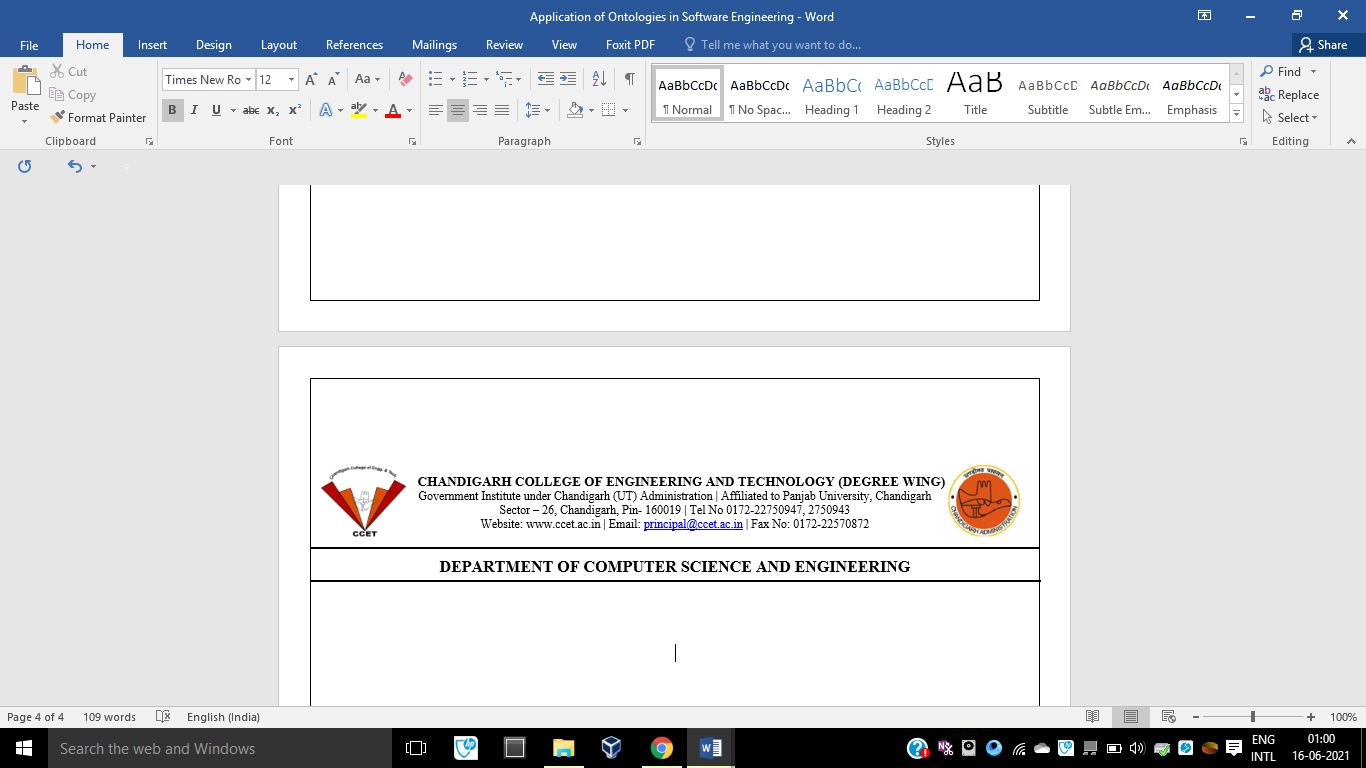
**CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(DEGREE WING)**

Government Institute under Chandigarh (UT) Administration, Affiliated to Panjab University, Chandigarh

Sector-26, Chandigarh. PIN-160019

**July, 2021**

**CHANDIGARH COLLEGE OF ENGINEERING AND TECHNOLOGY (DEGREE WING)**

Government Institute under Chandigarh (UT) Administration | Affiliated to Panjab University , Chandigarh

Sector-26, Chandigarh. PIN-160019 | Tel. No. 0172-2750947, 2750943

Website: www.ccet.ac.in | Email: principal@ccet.ac.in | Fax. No**. :**0172-2750872

**Department of Computer Sc. & Engineering**

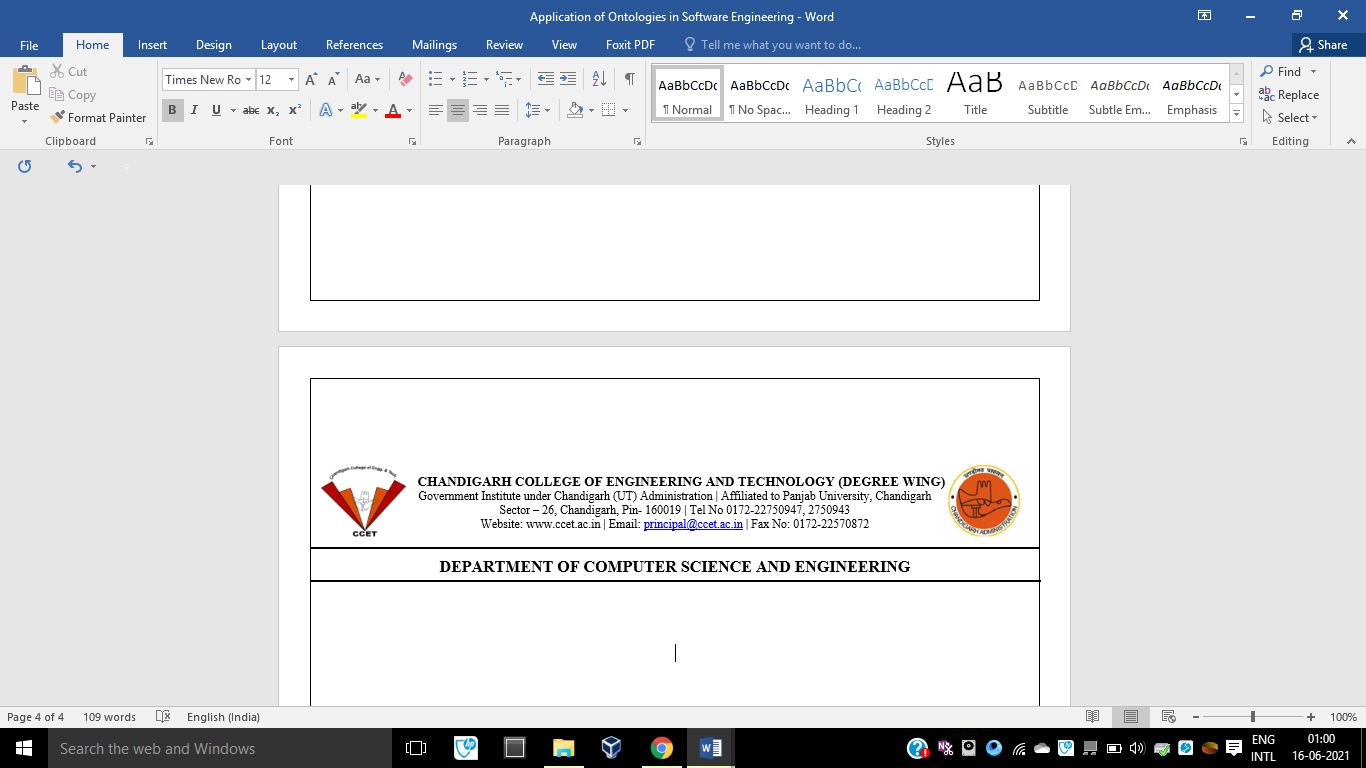
**CANDIDATE’S DECLARATION**

I hereby declare that the work presented in this report entitled “Student Login and Grievance System in MongoDB”, in fulfilment of the requirement for the award of the degree Bachelor of Engineering in Computer Science & Engineering, submitted in CSE Department, Chandigarh College of Engineering & Technology (Degree wing) affiliated to Punjab University, Chandigarh, is an authentic record of my/our own work carried out during my degree under the guidance of Dr. Dheerendra Singh (Professor in CSE, CCET).

The work reported in this has not been submitted by me for award of any other degree or diploma.

Date: 25 July, 2021 Taruna Saini

Place: CCET, Sector-26, Chandigarh CO19366



**CERTIFICATE**

This is to certify that the Project work entitled “Student Login and Grievance System in MongoDB” submitted by Taruna Saini in fulfilment for the requirements of the award of Bachelor of Engineering Degree in Computer Science & Engineering at Chandigarh College of Engineering and Technology (Degree Wing), Chandigarh is an authentic work carried out by him/her under my supervision and guidance.

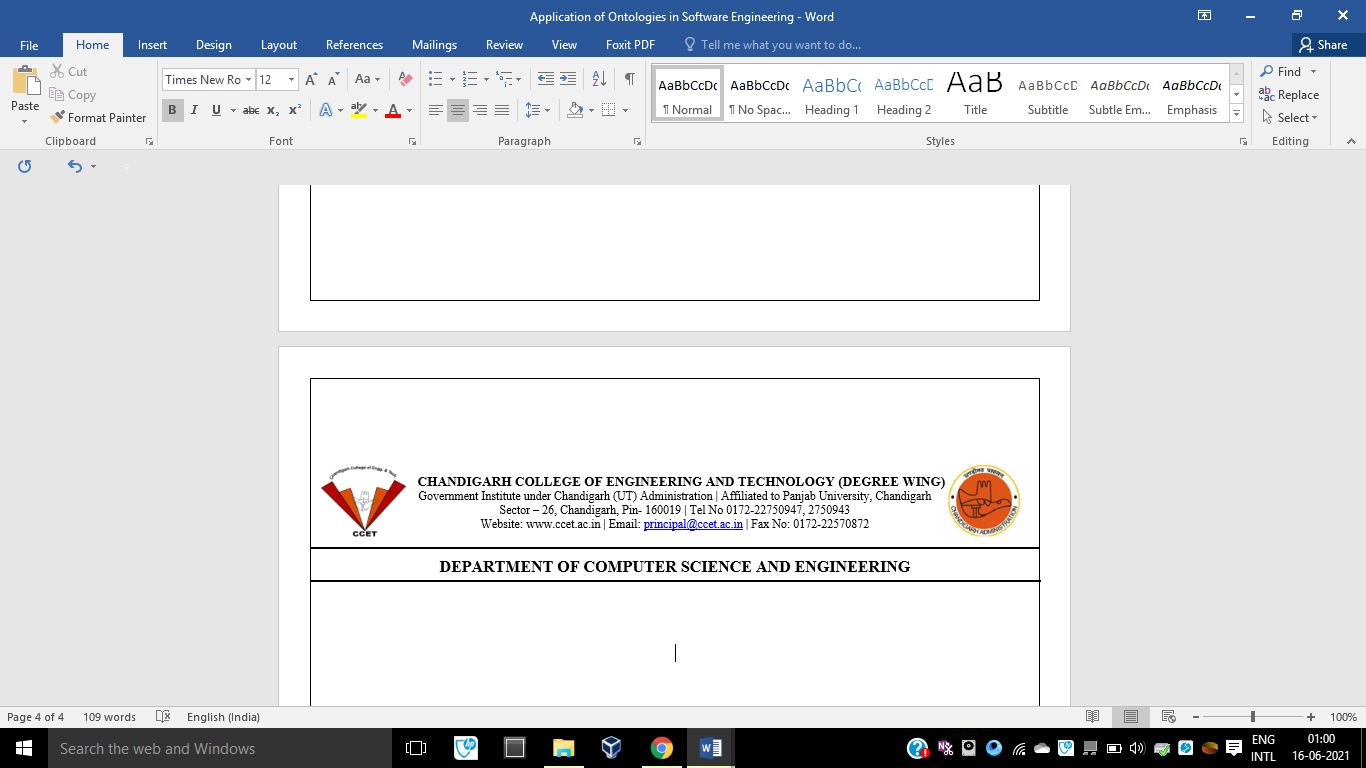
To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree.

Date: Dr. Dheerendra Singh

Place: CCET, Sector-26, Chandigarh Computer Science and Engineering

CCET (Degree Wing)

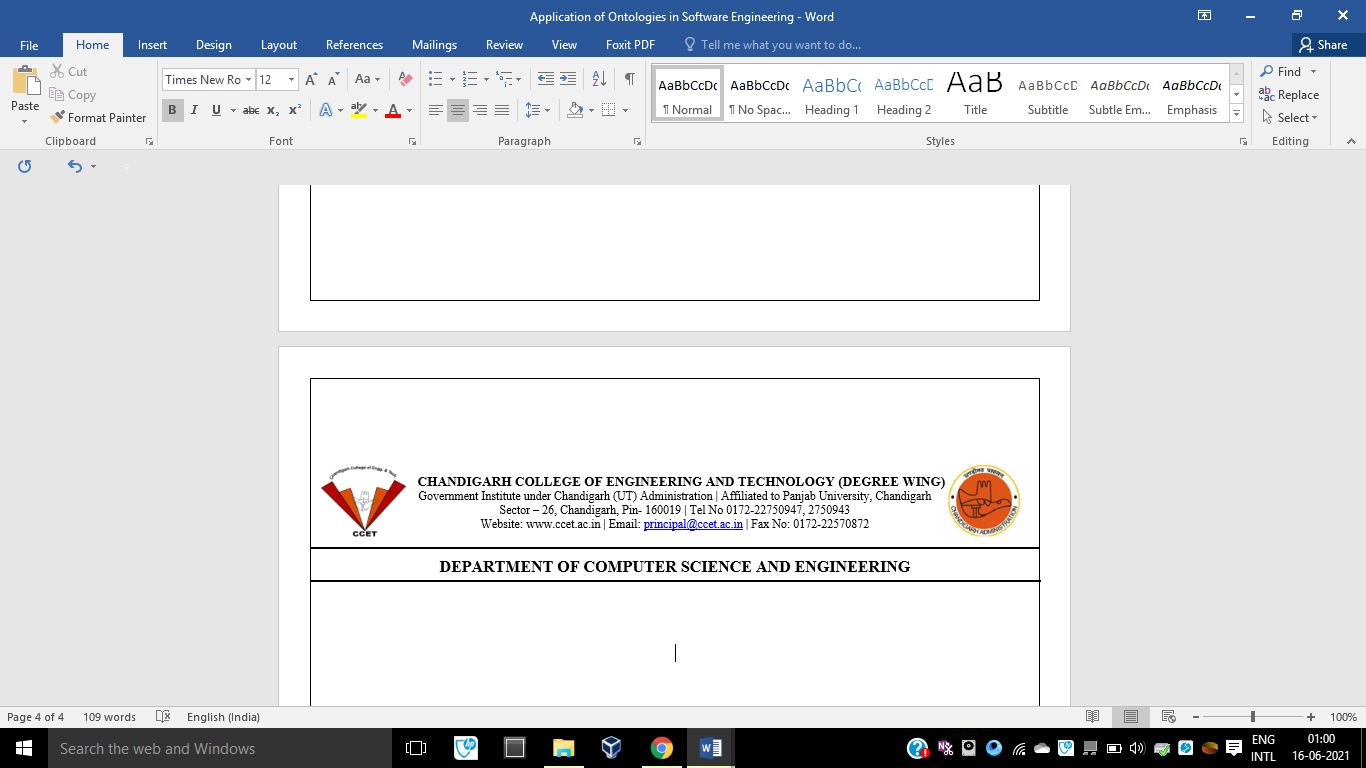
Chandigarh



**ACKNOWLEDGEMENT**

I am pleased to present “Student Login and Grievance System in MongoDB” and take this opportunity to express my profound gratitude to all those people who helped me in completion of this project. I thank my college for providing me with excellent facilities that helped me to complete and present this project. I would also like to thank the staff members and lab assistants for permitting me to use computers in the lab as and when required. I express my deepest gratitude towards my teacher, Dr. Dheerendra Singh for his valuable and timely advice during the various phases in my project. I would also like to thank him for providing me with all proper facilities and support as the project co-coordinator. I would like to thank him for support, patience and faith in my capabilities and for giving me flexibility in terms of working and reporting schedules.

I would like to thank all my friends for their smiles and friendship making the college life enjoyable and memorable and family members who always stood beside me and provided the utmost important moral support. Finally, I would like to thank everyone who has helped me directly or indirectly in this project.



**ABSTRACT**

The success of any organization such as a college or a university hinges on its ability to acquire accurate and timely data about its operations, to manage this data effectively, and to use it to analyse, improve and guide its activities. Integrated student database system offers students with a view to have a look at their details and to report any grievance to the authorities. To provide a single consistent result for every object represented in these data sources, data fusion is concerned with resolving data inconsistency present in the heterogeneous sources of data. The main objective of this project is to build a rigid and robust integrated student database system that will track and store records of students. This easy-to-use, integrated database application is geared towards reducing time spent on administrative tasks. The system is intended to accept process and generate report accurately and any user can access the system at any point in time provided internet facility is available. The system is also intended to provide better services to users, provide meaningful, consistent, and timely data and information and finally promotes efficiency by converting paper processes to electronic form. The system was developed using technologies such as MongoDB, Express JS, React JS and NodeJS. React JS is used to build the user interface and database was built using MongoDB. Node JS and Express JS were used to connect the UI to the database and served as a backend to the project.

**LIST OF TABLES**

**………………………………………………………………………………………………......**

1. Figure 1 – Kinds of Ontologies according to Generality level…………….…….8

**LIST OF FIGURES**

**………………………………………………………………………………………………......**

1. Figure 1 – React JS logo…………….…….8

**CONTENTS**

**………………………………………………………………………………………………......**

1. Student's Declaration…………………………………………………………..…...i
2. Certificate by the guide…………………………………………………………….ii
3. Acknowledgement..................................................................................................iii
4. Abstract...................................................................................................................iv
5. List of tables………………………………………………………………………..v
6. List of figures……………………………………………………………………...vi
7. Chapter 1- Introduction to MongoDB………………….........................................7
8. Definition……………………………………………..7
9. History…………………………………………………….……7
10. Features………………………………………………………………..
11. Why learn React JS?...............................................................................
12. Advantages………………………………………………………………
13. Disadvantages…………………………………………………………
14. React vs Angular………………………………………………………..
15. Chapter 2- Environment Setup………………………………........……8
16. Install NPM……………………………………………………………….
17. Install Node……………………………………………………………..
18. Create a react app……………………………………………………….
19. Clear out unwanted code………………………………………………
20. Uses of Ontologies in Software Engineering and Technology……....10
21. Scenarios for applying Ontologies………………………...…12
22. Ontology versus Conceptual Model………………………………….…...12
23. Ontology versus Metamodel…………………….…...13
24. Ontologies in Software Engineering Environments……………………........….13
    1. Mantis Environment……………………………….…….13
    2. TABA Workstation…………………………………….…..14
25. Representing Ontologies Using Software Engineering Techniques…..………..15
    1. REFSENO…………………………………………..15
26. Ontology Development Tools Based on Software Engineering Techniques…...16
    1. Protégé………………………………………………………....16
    2. DUET……………………………………………………….….16
    3. IBM Rational Rose UML Models……………………....…...16
    4. VOM…………………………………………………………...16
27. A proposal of taxonomy………………………………………………….…17
    1. Ontologies of Domain…………………………………….…..17
28. Advantages of Ontologies in Software Engineering……………………....……..19
29. Conclusion……………………………………………...……………..…20
30. References………………………………………………………...…………...21

**CHAPTER-1**

**INTRODUCTION TO MONGO DB**

1. **Overview**



Figure-1: Mongo DB logo

Mongo DB is an open-source document database and leading NoSQL database. MongoDB is written in C++.

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time

The document model maps to the objects in your application code, making data easy to work with.

1. **Advantages**

Any relational database has a typical schema design that shows number of tables and the relationship between these tables. While in MongoDB, there is no concept of relationship.

Advantages of MongoDB over RDBMS

* Schema less − MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
* Structure of a single object is clear.
* No complex joins.
* Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
* Tuning.
* Ease of scale-out − MongoDB is easy to scale.
* Conversion/mapping of application objects to database objects not needed.
* Uses internal memory for storing the (windowed) working set, enabling faster access of data.

Why Use MongoDB?

* Document Oriented Storage − Data is stored in the form of JSON style documents.
* Index on any attribute
* Replication and high availability
* Rich queries
* Fast in-place updates
* Professional support by MongoDB

Where to Use MongoDB?

* Big Data
* Content Management and Delivery
* Mobile and Social Infrastructure
* User Data Management
* Data Hub

1. **Installation**

To install MongoDB on Windows, first download the latest release of MongoDB from <https://www.mongodb.com/download-center>.

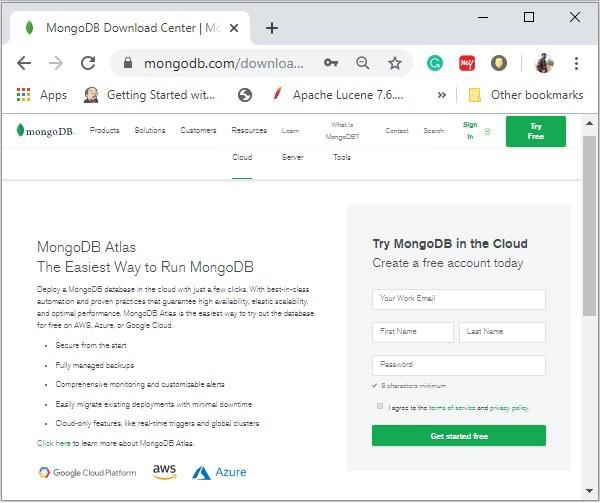


Figure-2: Downloading MonogDB

Enter the required details, select the Server tab, in it you can choose the version of MongoDB, operating system and, packaging as:

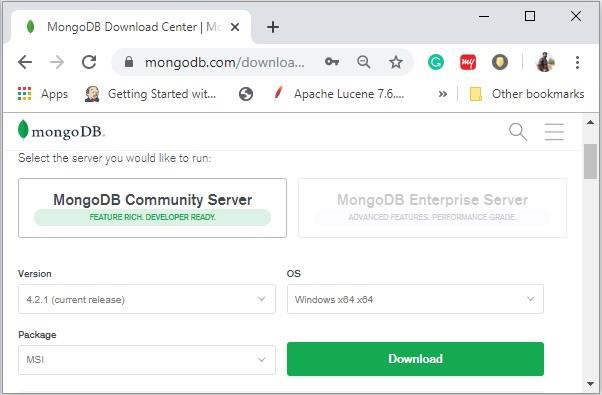


Figure-3: Downloading MongoDB

Now install the downloaded file, by default, it will be installed in the folder C:\Program Files\. MongoDB requires a data folder to store its files. The default location for the MongoDB data directory is c:\data\db. So you need to create this folder using the Command Prompt. Execute the following command sequence.

C:\>md data

C:\md data\db

Then you need to specify set the dbpath to the created directory in mongod.exe. For the same, issue the following commands. In the command prompt, navigate to the bin directory current in the MongoDB installation folder. Suppose my installation folder is C:\Program Files\MongoDB

C:\Users\XYZ>d:cd C:\Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongod.exe --dbpath "C:\data

This will show waiting for connections message on the console output, which indicates that the mongod.exe process is running successfully.

Now to run the MongoDB, you need to open another command prompt and issue the following command.

C:\Program Files\MongoDB\Server\4.2\bin>mongo.exe

MongoDB shell version v4.2.1

Connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("4260beda-f662-4cbe-9bc7-5c1f2242663c") }

MongoDB server version: 4.2.1 >

This will show that MongoDB is installed and run successfully. Next time when you run MongoDB, you need to issue only commands.

C:\Program Files\MongoDB\Server\4.2\bin>mongod.exe --dbpath "C:\data"

C:\Program Files\MongoDB\Server\4.2\bin>mongo.exe

1. **Data Modelling**

Data in MongoDB has a flexible schema.documents in the same collection. They do not need to have the same set of fields or structure Common fields in a collection’s documents may hold different types of data.

Data Model Design

MongoDB provides two types of data models: — Embedded data model and Normalized data model. Based on the requirement, you can use either of the models while preparing your document.

Embedded Data Model

In this model, you can have (embed) all the related data in a single document, it is also known as de-normalized data model.

Normalized Data Model

In this model, you can refer the sub documents in the original document, using references.

Considerations while designing Schema in MongoDB

* Design your schema according to user requirements.
* Combine objects into one document if you will use them together. Otherwise separate them (but make sure there should not be need of joins).
* Duplicate the data (but limited) because disk space is cheap as compare to compute time.
* Do joins while write, not on read.
* Optimize your schema for most frequent use cases.
* Do complex aggregation in the schema.

1. **MongoDB vs MySQL**

|  |  |
| --- | --- |
| **MongoDB** | **MYSQL** |
| MongoDB represents data as JSON documents. | MySQL represents data in tables and rows. |
| In MongoDB, you don't need to define the schema. Instead, you just drop in documents don't even need to have the same fields. | MySQL requires you to define your tables and columns before you can store anything, and every row in a table must have the same columns. |
| MongoDB has a pre-defined structure that can be defined and adhered to, but also, if you need different documents in a collection, it can have different structures. | MySQL uses Structured Query Language (SQL) for database access. You can't change the schema. |
| Supported languages are C++, C | Supported languages are C++, C and JavaScript |
| Ongoing development is done by MongoDB, Inc. | Constant development is done by the Oracle Corporation. |
| MongoDB supports built-in replication, sharding, and auto-elections. | MySQL supports master-slave replication and master replication. |
| If an index is not found, every document within a collection must be scanned to select the documents which offer a match to the query statement. | If an index is not defined, then the database engine needs to scan the complete table to find all relevant rows. |
| GPL v2/ Commercial license available OD | GNU AGPL v3.0/ Commercial licenses available OD |
| If most of your services are cloud based MongoDB is the best suited for you. | If data security is your priority then MYSQL is the best option for you. |
| MongoDB places no restrictions on schema design. | MySQL requires you to define your tables and columns before you can store anything. Every row in a table must have the same columns. |
| MongoDB uses JavaScript as query language. | MySQL uses the Structured Query Language (SQL). |
| MongoDB doesn't support JOIN. | MySQL supports JOIN operations. |
| It has the ability to handle large unstructured data | MySQL is quite slow in comparison to MongoDB while dealing with large databases. |
| Real-time analytics, content management, internet of things, mobile apps | Structured data with clear schema |
| No schema definition required so lesser risk of attack due to design | Risk of SQL injection attacks |
| An ideal choice if you have unstructured and/or structured data with the potential for rapid growth. | A great choice if you have structured data and need a traditional relational database. |

Table-1: MongoDB vs MySQL

KEY DIFFERENCES:

* MongoDB represents data as of JSON documents whereas MySQL represents data in tables and rows.
* In MongoDB, you don't need to define the schema while in MySQL you need to define your tables and columns
* MongoDB doesn't support JOIN but MySQL supports JOIN operations.
* MongoDB uses JavaScript as query language while MySQL uses the Structured Query Language (SQL).
* MongoDB is an ideal choice if you have unstructured and/or structured data with the potential for rapid growth while MYSQL is a great choice if you have structured data and need a traditional relational database.
* If most of your services are cloud based MongoDB is the best suited for you but If data security is your priority then MYSQL is the best option for you.

**CHAPTER 2**

**MONGO DB COMMANDS**

1. **Use database command**

MongoDB use DATABASE\_NAME is used to create database. The command will create a new database if it doesn't exist, otherwise it will return the existing database.

Syntax

Basic syntax of use DATABASE statement is as follows –

use DATABASE\_NAME

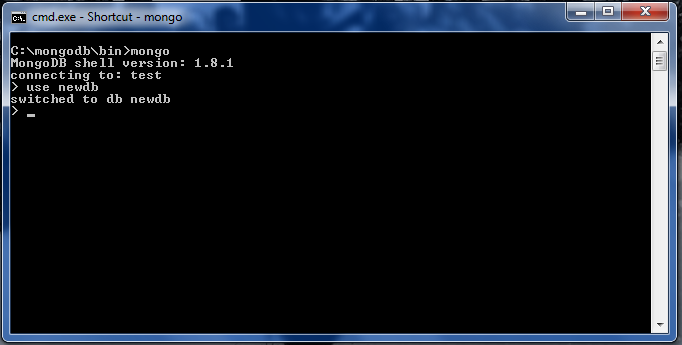


Figure 4 – use command in MongoDB

1. **dropDatabase() command**

MongoDB db.dropDatabase() command is used to drop a existing database.

Syntax

Basic syntax of dropDatabase() command is as follows –

db.dropDatabase()

This will delete the selected database. If you have not selected any database, then it will delete default 'test' database.

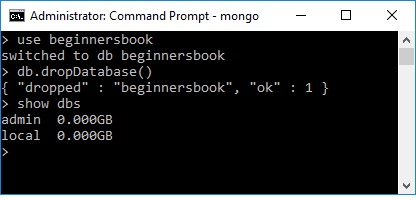


Figure 5 – dropDatabase() command in MongoDB

1. **createCollection() method**

MongoDB db.createCollection(name, options) is used to create collection.

Syntax

Basic syntax of createCollection() command is as follows –

db.createCollection(name, options)

In the command, name is name of collection to be created. Options is a document and is used to specify configuration of collection.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| Name | String | Name of the collection to be created |
| Options | Document | (Optional) Specify options about memory size and indexing |

Figure 6 – Syntax of createCollection() method

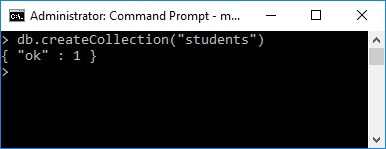


Figure 7 – createCollection() in MongoDB

1. **collection.drop() method**

MongoDB's db.collection.drop() is used to drop a collection from the database.

Syntax

Basic syntax of drop() command is as follows −

db.COLLECTION\_NAME.drop()

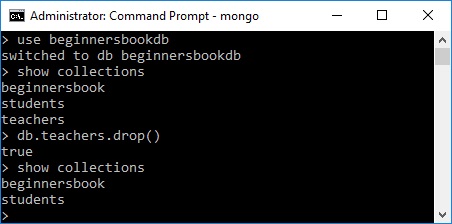


Figure 8 –db.collection.drop() method in MongoDB

drop() method will return true, if the selected collection is dropped successfully, otherwise it will return false.

1. **insert() method**

To insert data into MongoDB collection, you need to use MongoDB’s insert() or save() method.

Syntax

The basic syntax of insert() command is as follows –

>db.COLLECTION\_NAME.insert(document)

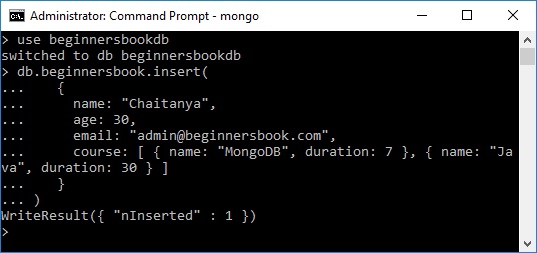


Figure 9 –insert() method in MongoDB

**CHAPTER -3**

**PROJECT**

1. **Project Introduction**

Managing Student databases for a university/college/school administration is greatly simplified and made effective with the advent of digital database systems such as MongoDB, MySQL, etc. instead of the original tradition of storing data in files which was practised years ago. Students can even access their records sitting in the comfort of their homes and can make or request changes whenever needed.

So, in this project; I have created a website project for managing the student database of my college. The website is made using MongoDB, React JS, Express JS and Node JS. MongoDB is used as a database management system and takes input from the front-end system and stores all the entries in the collection.

The project is made to make the management of student database easier and less time-consuming. Anyone with the request access can access the database from any part of the world since the database was uploaded on MongoDB Atlas which makes this possible.

The students can login once they are registered / signed in and then they can view their details or make a grievance complaint. The website fetches real-time dynamic content once the details are registered and to make things easier and more pleasing for the student, the stored details are fetched onto the front-end React JS part too.

The website is used to store any grievance the student makes and also stores the date in which the grievance was made. The administration can then sort out the messages (grievances) according to the dates by running a simple query. The messages are stored in the same document (under the same name) if a user writes more than one grievance.

Hence, this student database system is used to login a student or register the student into the database. After login/ registration, the student can view his/her details and also send a grievance message to the authorities who receive all the data via MongoDB.

1. **Project Code Components**

The project contains two folders – client and server.

Client folder –

The client folder contains the front-end portion of the project. This part contains or focuses on the visible section which is seen by the client. Hence this folder is named as the client folder. The client folder is a react app and is built out of React JS. It can be run on the local machine using the command – *npm start.*

Here are the contents of the client folder –

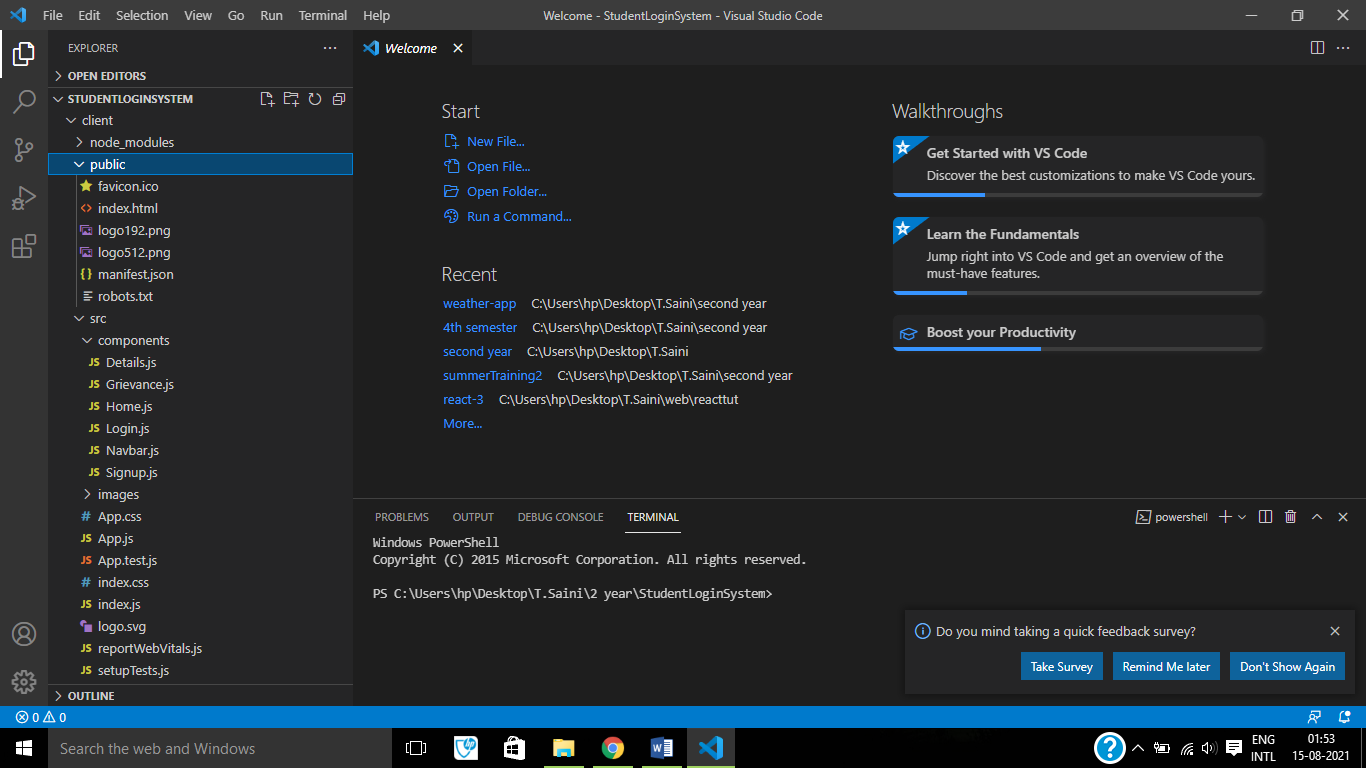
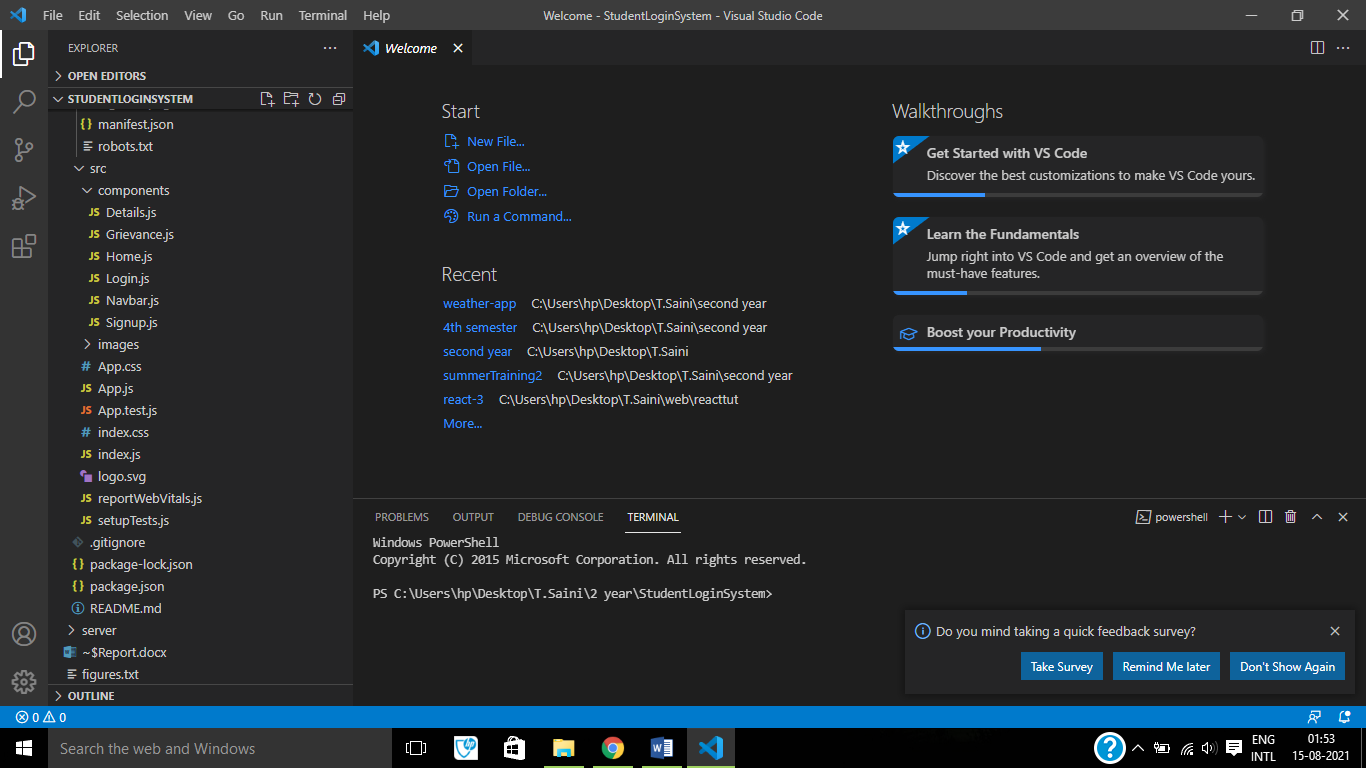


Figure 10 - Contents of the client folder

Server folder –

The server folder is the back-end portion of the project. It handles all the part which isn’t visible to the client. The server is based up on languages of Node JS and Express JS. These languages are connected to the database i.e. MonogDB. The server can be run with the command –

*node app.js*

The server runs on the port 5000 on the local machine.

The contents of the server folder are –

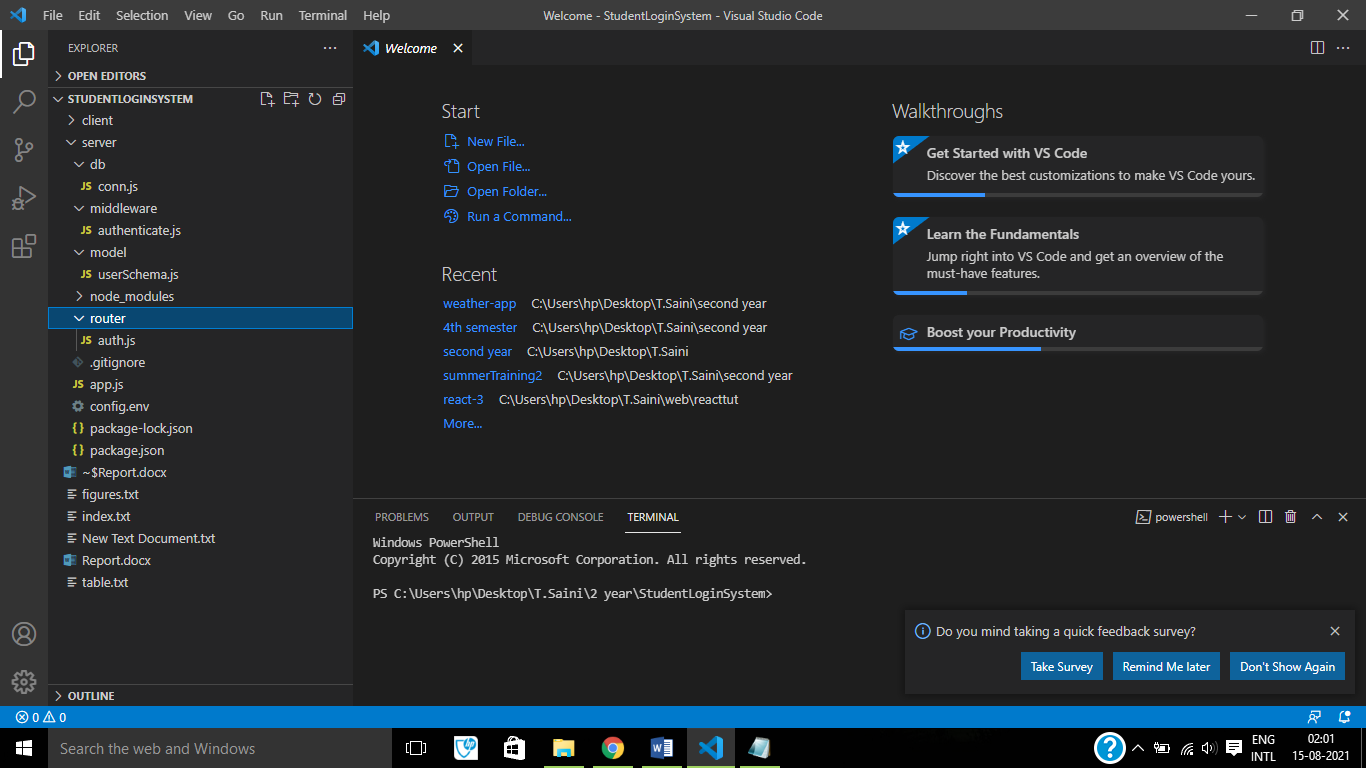


Figure 11 - Contents of the server folder