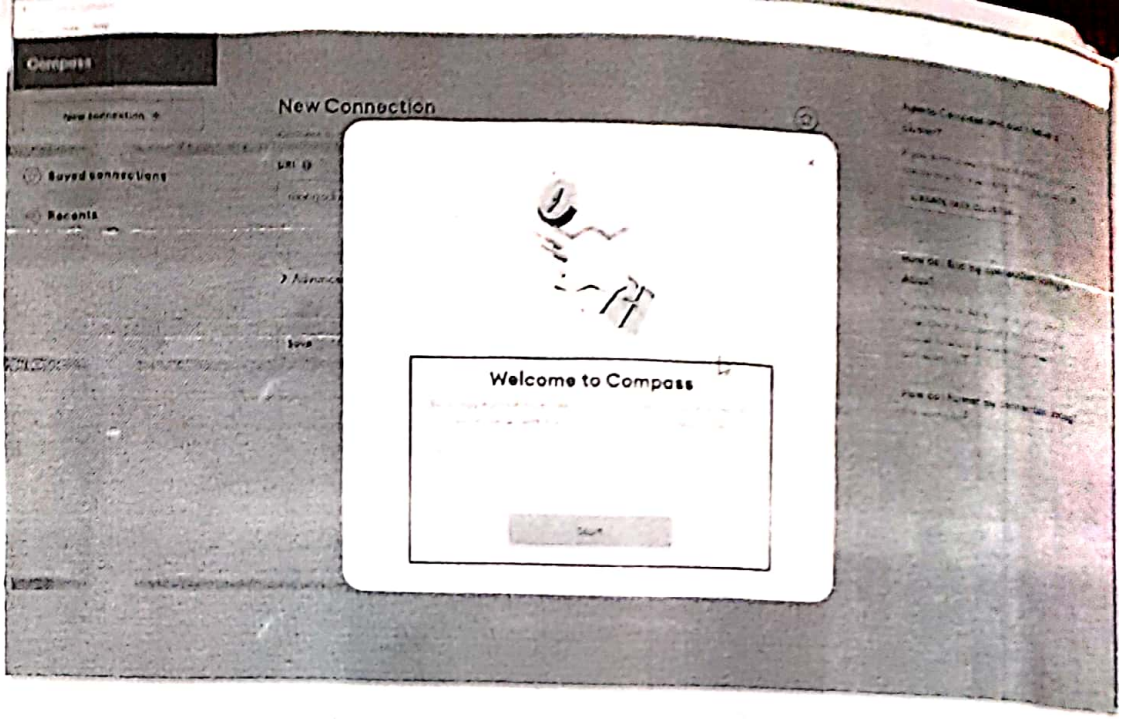
INDEX

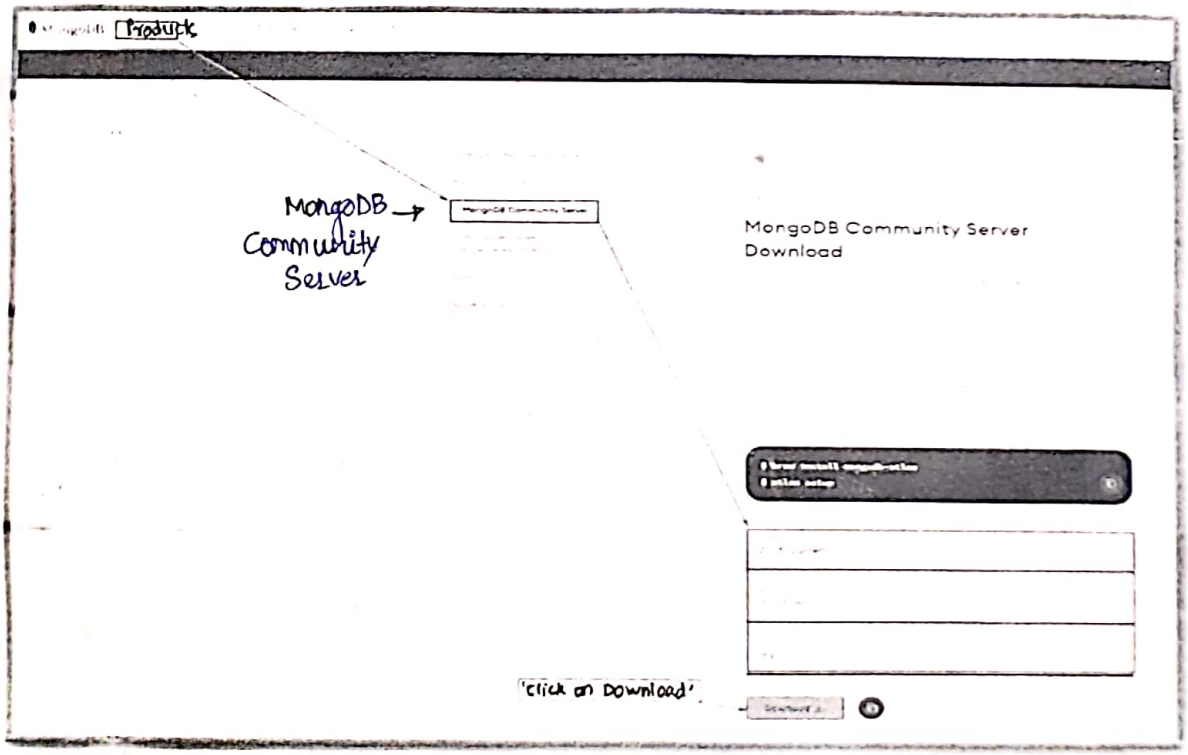
S. No.	Title of Skill Activity	Date of Allocation	Date of Submission	Sign of Faculty
01.	Creating a Database by setting MONGODB ATLAS Cloud and Installing MONGODB Compass.	08/02/2023	15/02/2023	
02.	Implementing CRUD Operation. Illustration of CREATE and READ command of MONGODB.	15/02/2023	01/03/2023	
03.	Implementing CRUD Operation. Illustration of UPDATE and DELETE command of MONGODB.	15/03/2023	22/03/2023	
04.	Designing a NoSQL Database for a Project.	23/03/2023	12/04/2023	
05.	Implementation of a Database Management Project using MONGODB as Backend.	19/04/2023	29/04/2023	

Step-01:

(a) Go to Mongo Compass

(b) Click on Next & select 'compass'





Step-02:

(a) Under 'Products' → Select 'MongoDB Community Server'

↳ Download for the "Windows"

Download it.

Step-03: (a) To check the version of (mongo-db, use command prompt. ↳shell)

(b) Put 'mongo' cmnd, & all the details of MongoDB Server.

```
C:\Users\Vimal Dagar>mongo
MongoDB shell version v4.4.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fdb58786-ab40-4ae6-8bc8-f2da10194453") }
MongoDB server version: 4.4.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2021-04-28T11:32:41.533+05:30: Access control is not enabled for the database. Read and
data and configuration is unrestricted
```

SKILL ACTIVITY NO: 1

Date: 08/02/2023

(To be filled by the Student)

Title:

Creating a Database by setting MongoDB Atlas Cloud and installing MongoDB Compass.

1. What is the purpose of this activity ? (Explain in 3 – 4 lines)

The purpose of this activity are as follows:- (To understand

(i) What are NoSQL databases? Why we need them?

(ii) Implementation of NoSQL databases, open-source NoSQL db's.

(iii) (What, Why, How) of MongoDB NoSQL database.

(iv) Various features of Atlas Cloud {Mongodb as a service}

(v) Installation of MongoDB Compass on our local system.

(vi) Query Engine & Document-type data storage models. → why they are better than SQL? → what's the Real-world usecase of them?
Working of → Algo's working behind the documentdb's CRUD operation.

2. Step performed in this activity (Explain in 5 – 6 lines)

Steps performed in this activity are as follows:-

Installing MongoDB server.

Step (i) Open your Browser & browse "MongoDB download" → URL: https://www.mongodb.com/try/download/community

Step(ii) Click on "MongoDB Community Server Download".

Step(iii) Under Products, go to MongoDB Comm. Server & select → Version: 4.4.5 LTS, Platform: Windows, Package: msi (As per user's requirement & specifications of machine)

Step(iv) Click on "Download" & simultaneously download MongoDB shell & MongoDB compass.

Step(v) Downloading MongoDB Compass: go to following URL: https://www.mongodb.com/products/compass

Step(vi) After downloading both Server & compass :-
- Double-click on ".msi" Installer → & Next → choose "Custom"
- Under Custom Installation → Select "MongoDB as a service" → Run it as Network service User.
- Then, click on "Next"

Step(vii) Also, Edit the Environment Variables → So, paste the binaries path → i.e. "C:\Program Files\MongoDB\Server\4.4\bin"

3. What resources / materials / equipments / tools did you use for this activity ?

1. Online Resources :
   - MongoDB Community Server V4.4.5
2. (https://mongodb.com/try/download/community)
   - MongoDB official documentation
3. (https://www.mongodb.com/docs/atlas)
4. - MongoDB Compass documentation
   - (https://mongodb.com/download-center/compass)
5. Equipments : PC : Intel(R) Core (TM) i5-6400@2.70MHz
   LENOVO_MT_30_AS_BU
   LENOVO_FM_ThinkStation P320
   - Microsoft Windows 10 Pro 10.0.19044 Build
   - Greenshot Screenshot Tool
6.
7. Materials :
   - Lecture PPT on Installation of MongoDB Community Servers.
8.

4. What skills did you acquire ?

1. W5HH abt NoSQL databases
2. W5HH abt Document-type databases
3. How, queries evaluation plan, algo's created by developers of MongoDB?
4.
5. Real-world use cases of MongoDB NoSQL database
6. Installation of "MongoDB Compass" & "MongoDB Server Community Server"
7.
8. Database creation on MongoDB Server & connecting with MongoDB Compass.

5. Time taken to complete the activity ? 02:00 (hours)

Signature of Student

(To be filled by Faculty)

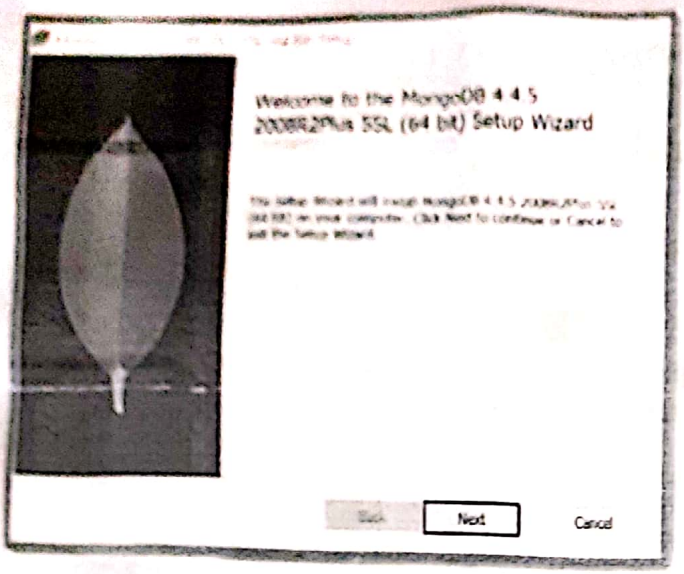
S. No.	Skills / Competencies	(Achieved / Not Achieved) (YES / NO)
01.	Installation of MongoDB Community Server (V6.0.5)	
02.	Installation of MongoDB Compass in our local PC. (V6.0)	
03.	(Why, What, How) regarding to NoSQL databases	
04.	Provisioning the MongoDB Compass in the cloud using ATLAS,	
05.	Realworld usecases of MongoDB database as DOCUMENT BASED Database.	

Remarks

Total marks _____ out of 10.

Sign of Faculty

Date:

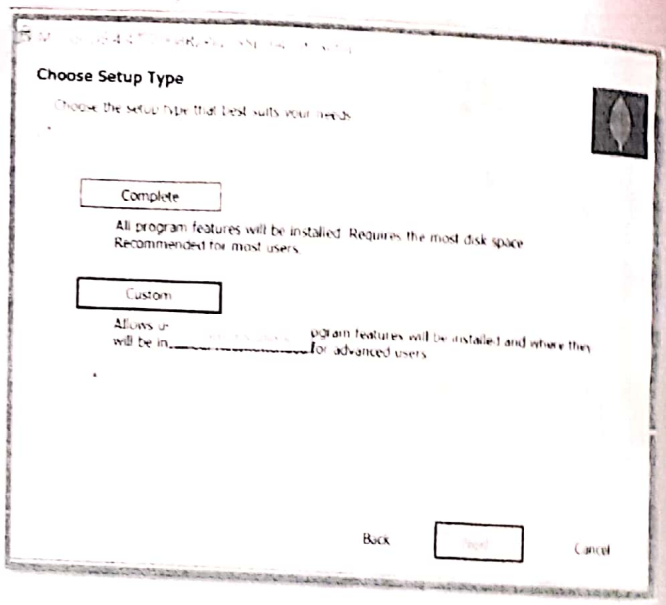


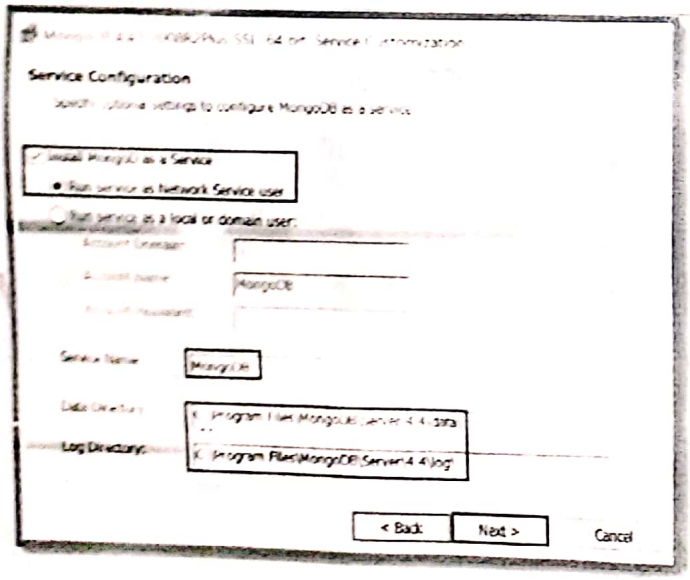
Step-04: After downloading setup, double-click on MongoDB Installer (V4.4.5).

↳ Click on 'Next' Button.

Step-05: Now, select 'custom' setup type,

↳ Here, we can easily customize MongoDB Community server as a Thread (Process) and using cloud paradigm of 'MongoDB as a service'.





Step-06: Under 'Service Configuration',

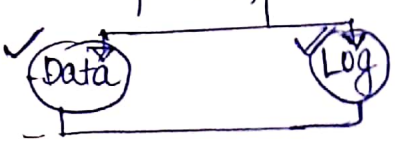
Select to

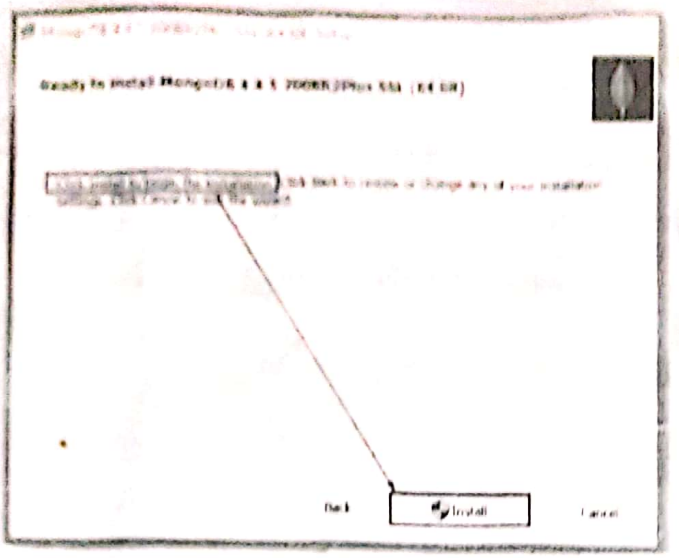
a) ☑ Install MongoD as a Service (Mgdaas)

↳ Run service as Network Service User

b) Select 'Service Name' == MongoDB

c) Provide desired path of directory:-



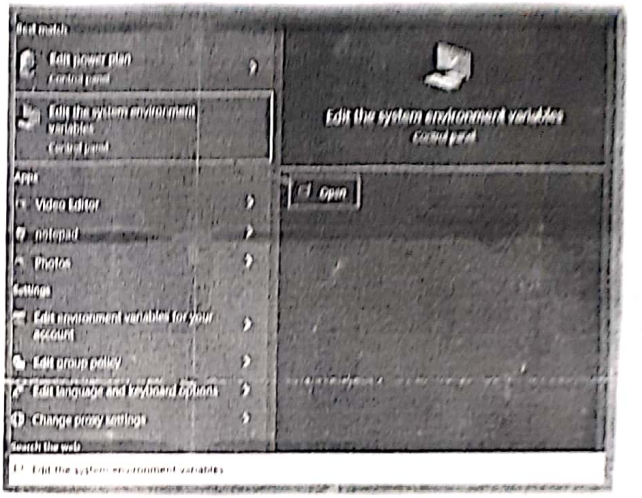


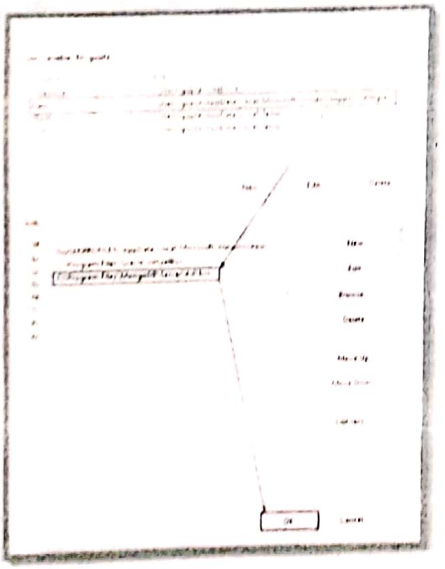
Step-07: Click on the mongodb 4.4.5 2008R2 (Install) → And it will start installing the server.
(by default, Accounts (control) (prompting) (give the permissions)

Step-08: Now, to use mongo shell command from normal user cmnd prompt :-

(a) Search for Edit environmental variables, & open it.

(b) After opening it, search for "System Environmental Variables."



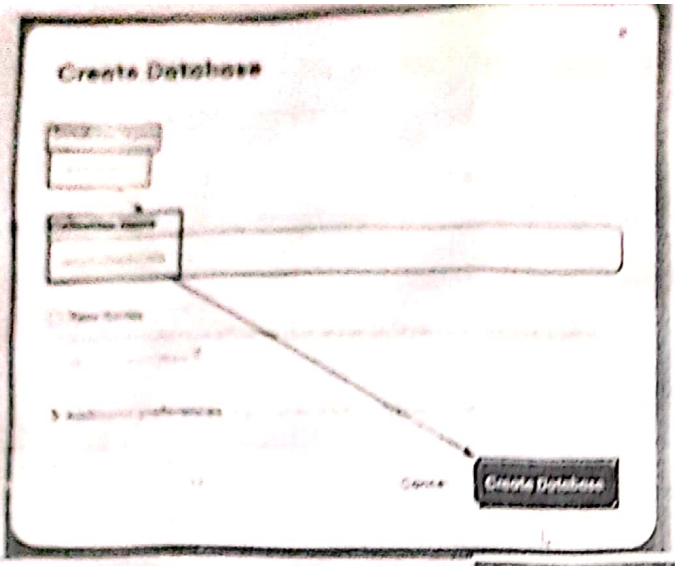


Step-09: Under "System Environmental Variables",

(a) Click on "New" button.

(b) Enter the 'binaries' → 'bin' path of MongoDB Server (Community) i.e.

(c) Then click on "OK" button. Due to which Environmental variable.



Step 01: CRUD Operation: CREATE

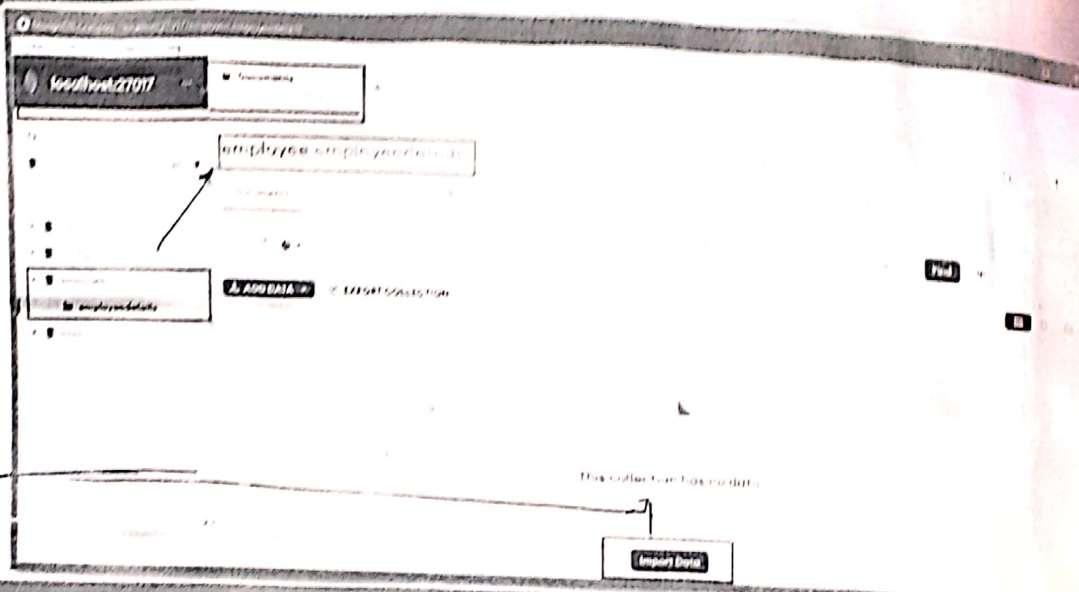
(a) In this screenshot, we are creating a database 'employee' => DATABASE_NAME

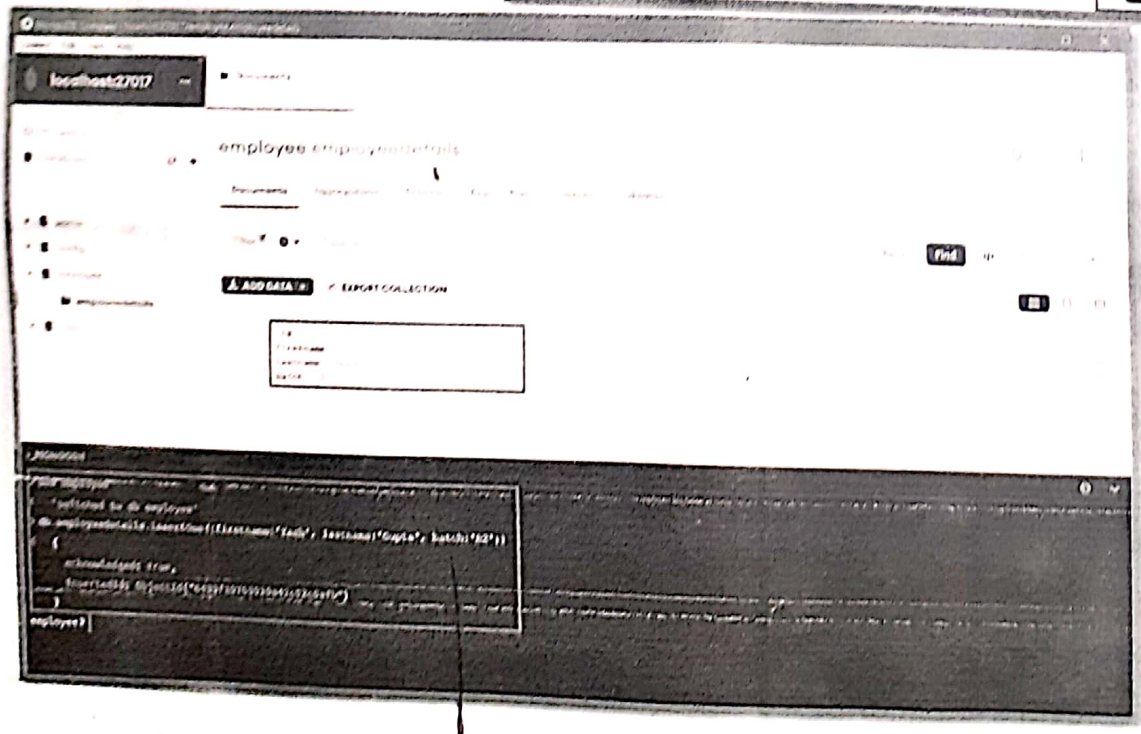
(b) Also, providing COLLECTION_NAME = 'employeedetails'

(c) Finally, click on 'Create Database'

Step 02: In this screenshot, the employee.employeedetails database & collection Both are (Created) successfully.

And we will insert the data Now.





Step 03: In this screenshot, Using Mongoshell,

(a) select the db

cmnd: use employee

(b) For INSERTing

(I) Insert data or Row in our database.

Use cmnd:

db.employeedetails.

insertone.({ firstname: 'Y
lastname: 'G
batch: '82
})

SKILL ACTIVITY NO: 2 Date: 16/02/2023

(To be filled by the Student)

Title:

Implementing CRUD operation : Illustration of CREATE & READ command of MongoDB.

1. What is the purpose of this activity ? (Explain in 3 – 4 lines)

The purpose of this activity are as follows:-

(i) To understand what are CRUD Operations in a NoSQL database?

(ii) To understand how these CRUD operations performed are different from the operations performed in SQL databases?

(iii) To understand & implement the CREATE & READ Operations in a NoSQL database i.e. MongoDB Community Server (V4.4.5.)

(iv) Real-World application → via creating a Database, collection & Inserting data fields (i.e. Rows/Tuples) in DocumentBasedDB.

2. Step performed in this activity (Explain in 5 – 6 lines)

Steps performed in this activity are as follows:-

Step(i): Open MongoDB Compass. i.e. { if installed locally, → then in PC use MongoDB compass Installer }

Step(ii): Click on the button → 'Create Database'

Step(iii) After the dialogue popup → Enter following details such as { → Database Name : employee, → Collection Name : employee Details }

Step(iv) Then, as per our database schema → we create our DB → { employee → employeedetails } 'document (Table)'

Step(v) After, succesfully completing step4 → we will CREATE the and insert our first Record. using Mongo shell { → already installed }

Step(vi) Then, we execute the CRUD i.e. CREATE and READ

3. What resources / materials / equipments / tools did you use for this activity ?

1. Online Resources
   ↳ MongoDB Community Server V4.4.5
2. (https://mongodb.com/try/download/community).
3. ↳ MongoDB Official documentation
   (https://www.mongodb.com/docs/atlas)
4. ↳ MongoDB Compass Download.
   ↳ (https://mongodb.com/download-center/compass)
5. Equipments: PC: Intel(R) Core(TM) i5-6400@2.70GHz
   LENOVO_MT_30_AS_BU
   LENOVO_FM_ThinkStation P310
   • Microsoft Windows 10 Pro 10.0.19044 Build
   • Greenshot Screenshot
7. Materials:
   ↳ Lecture PPT for Installation of MongoDB Compass.
8.

4. What skills did you acquire ?

1. W5HH abt CREATE & READ operations
2. Query Evaluation Engine of MongoDB Database
3. Implementation of MongoDB Compass on Local System.
4. Implementation of MongoDB shell
5. Comparison between CREATE and READ of [NoSQL database & SQL database]
6.
7.
8.

5. Time taken to complete the activity ? 02:00 (hours)

Signature of Student

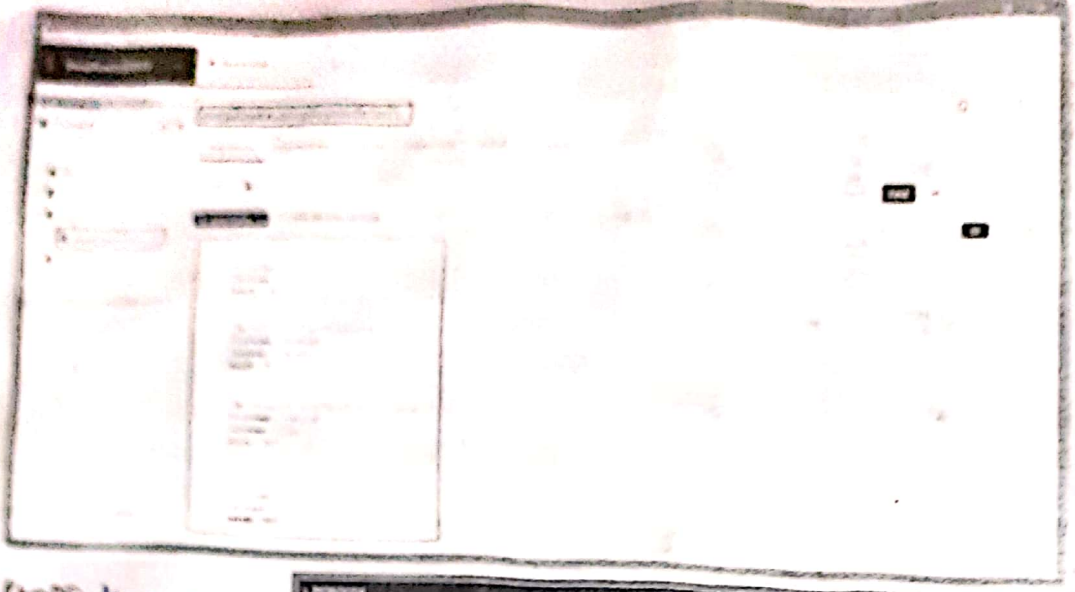
(To be filled by Faculty)

S. No.	Skills / Competencies	(Achieved / Not Achieved) (YES / NO)
01.	Performing CRUD operations (i.e. CREATE and READ) operations on the MongoDB community server v8.0.	
02.	Implementing and Creating MongoDB Compass on local PC via inserting the Records = docs in the DB	
03.	Also, implementing MongoDB shell & executing the commands.	

Remarks

Total marks _____ out of 10.

Sign of Faculty

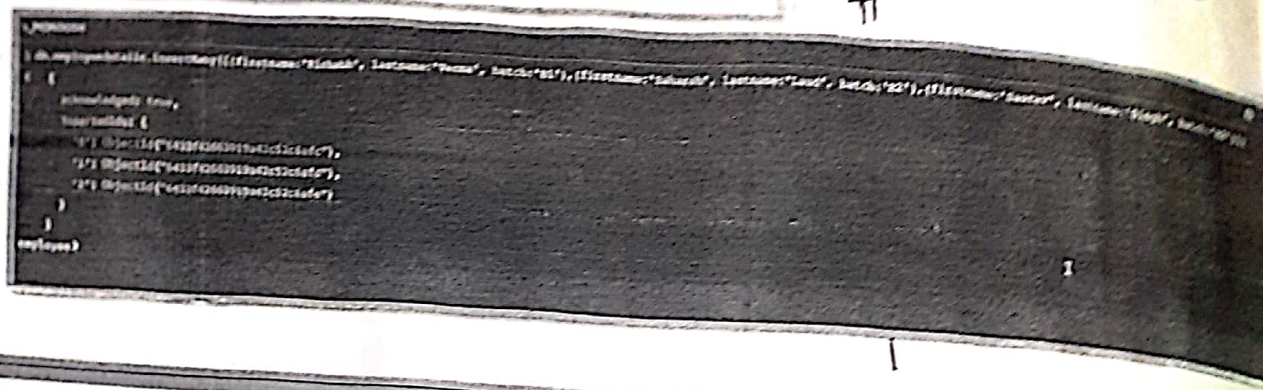


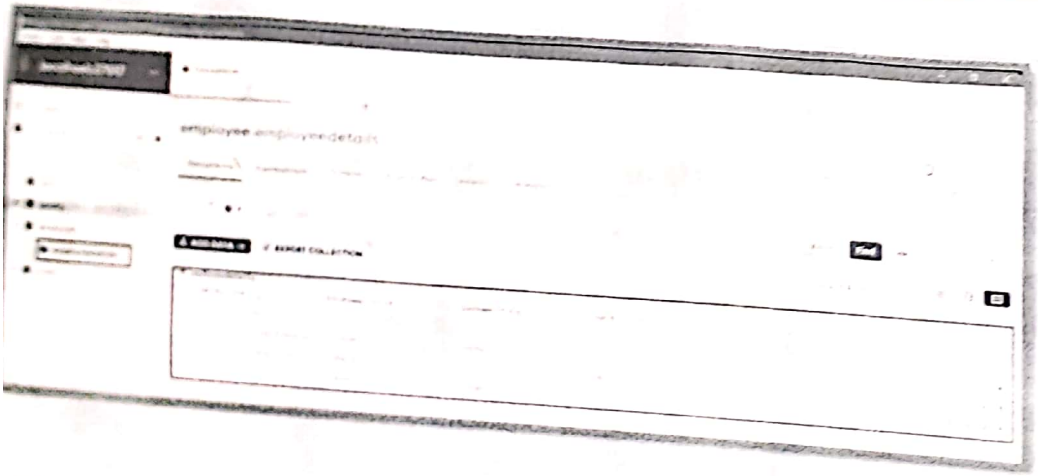
Step 04: In this screenshot, we can see the Datafields & records/tuples are inserted i.e. data has been INSERTED into the database.

Format of DB-data: JSON

Func: | .insertMany([ ]) |

Step 05: Here, we are using Mongoshell, to Insert Many records at a given time.





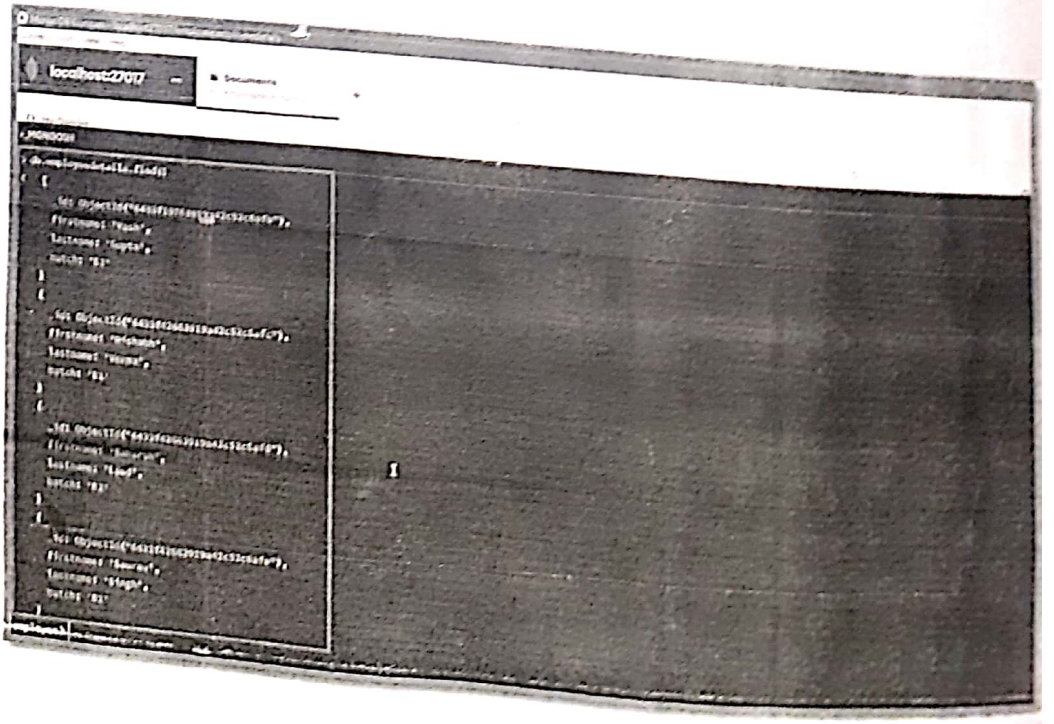
Step 06: Here, we are just changing the view of database.

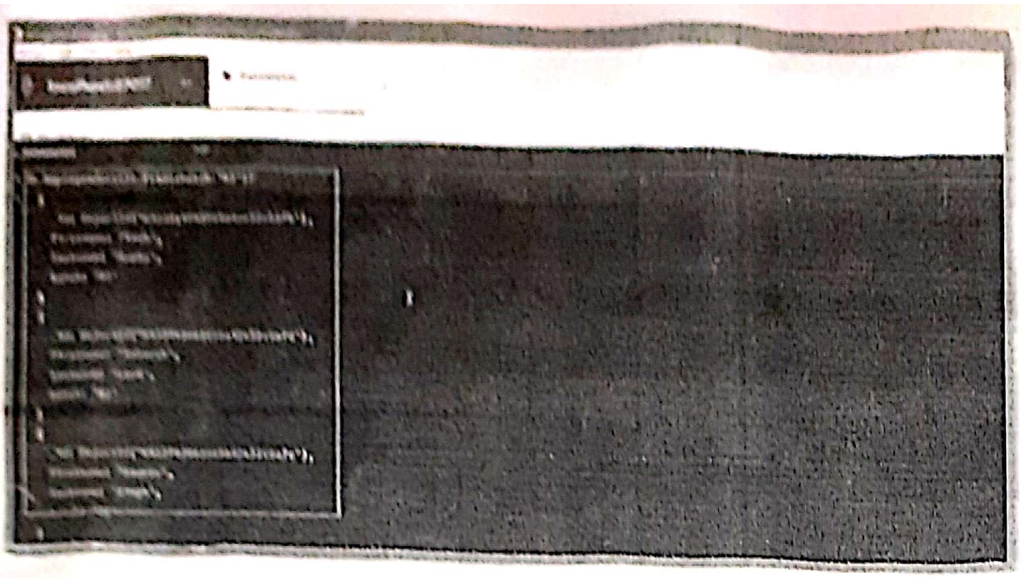
Means: → displaying DOCUMENTS in a tabular view as provided by MongoDB compass.

Step 07: In this screenshot, we are performing READ operation using the inbuilt function of MongoDB i.e. |.find()|.

↳ After, executing the command in Mongoshell,

↳ we will get all the (4 Objects = 4 Doc's = 4 Records)



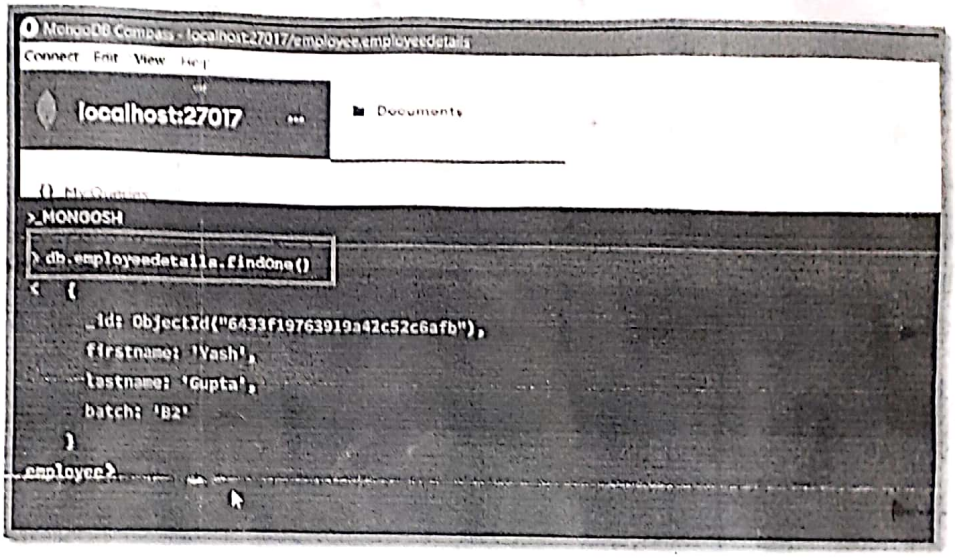


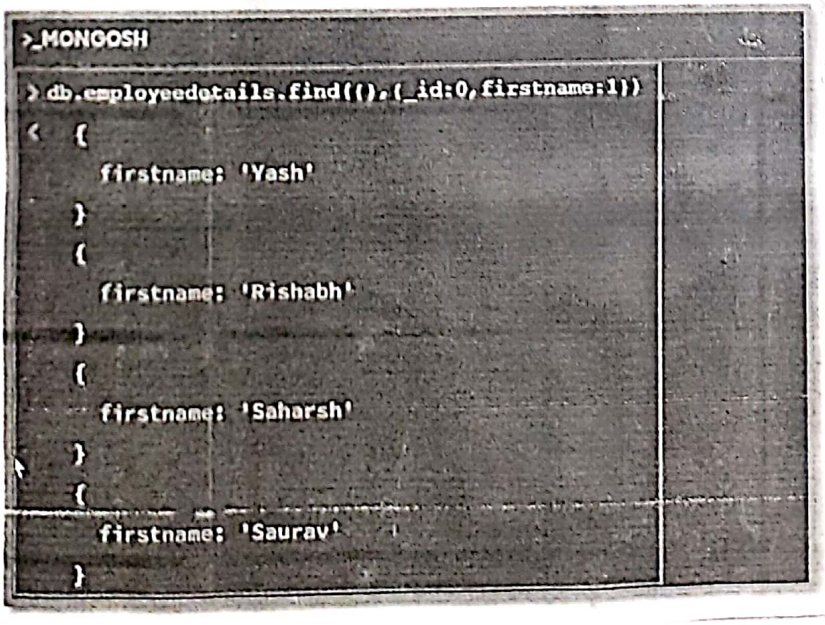
Step-08: Here, we are querying to find all those Records = Rows which have 'BATCH: 'B2'' (batch)

{batch of B2} → & as an output we get all those Records.

Step-09: In this screenshot, we are using Inbuilt functionality of function of MongoDB i.e. .findOne()

↳ O/P: As an output, we will get the FIRST {_id:0} Record of the Table.

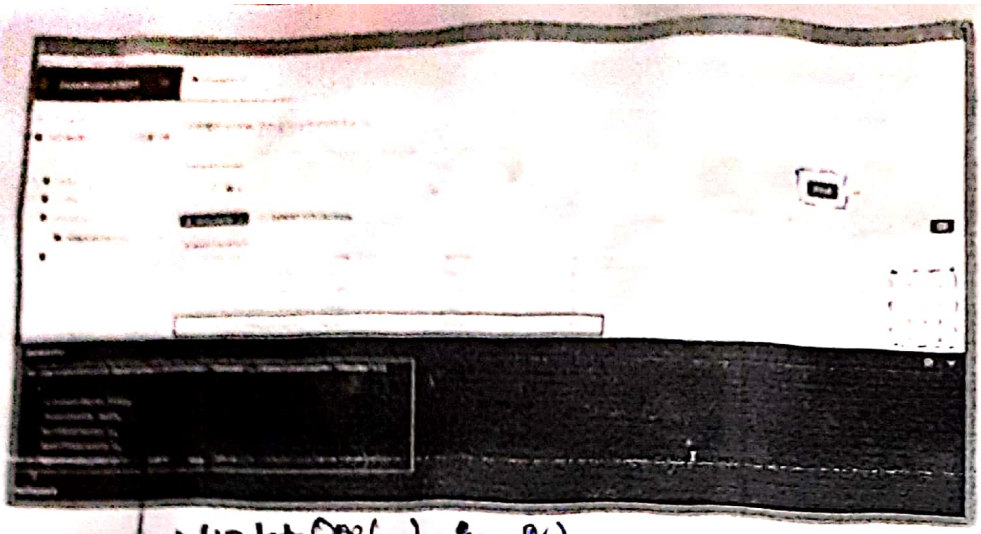




Step-10: In this screenshot, we are querying the database to find all those (Records / Rows / Tuples) where

↳ _id:0, } → &
↳ firstname: 1

↳ as an output, we are getting all the 'firstname:' ↳, ↳ using (.find())



Step_01: In this screenshot, as our database employee is already created. Also, the data tuples are stored.

↳ Now, we want to update SINGLE RECORD

↳ Therefore; mongodb cloud shell provide a function: .updateOne({ }) — { Here, enter the data which we want to update.

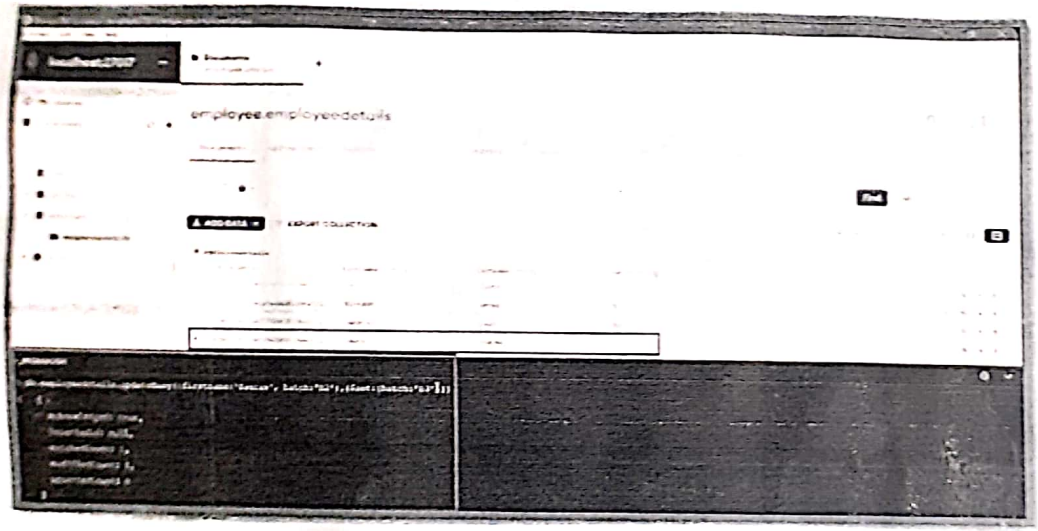
↳ updateOne( ). func^n:)

↳ CRUD

↳ UPDATE operation

↳ but only on ONE Record. ;)

i.e. (updating the details of one Record)



Step_02:

↳ In this screenshot, we are updating

↳ Many data fields / rows / tuples in one (single command)

↳ Mango-DB has provided one Inbuilt function i.e. .updateMany({ })

CRUD

(UPDATE Operation)

Here, we are updating all the record where { 'firstname: Saurav' & (AND) 'batch: B2' }

These set the 'batch: B3' ✓ & update it in the database.

SKILL ACTIVITY NO: 3

(To be filled by the Student)

Date: 15/03/2023

Title:

Implementing CRUD Operation : Illustration of UPDATE & DELETE command of MongoDB

1. What is the purpose of this activity ? (Explain in 3 – 4 lines)

The purpose of this activity are as follows:-

(i) To understand what are CRUD operations in MongoDB (document based DBS).

(ii) To understand how these CRUD Operations perform as compared to SQL database? & Complexity wise analysis → using perf_schema db given in MongoDB.

(iii) To understand and implement the UPDATE & DELETE operations in a NoSQL database i.e. MongoDB Community Server v4.4.5.

(iv) Real-world application :→ via creating & Updating & Deleting the rows/tuples in MongoDB. database engine.

2. Step performed in this activity (Explain in 5 – 6 lines)

Steps performed in this activity are as follows:-

Step(i) From previous skill activity we already launched the "DB Server", created the collection name. Now, we are using CRUD operation i.e. UPDATE & DELETE

Step(ii) For UPDATE Operation in Database

- (a) Execute .updateONE( ) & as argument → use($set.) func^n
- (b) Execute .updateMany( ) & as argument → use ($set.) func^n

Step(iii) For DELETE Operation in Database.

- (a) Execute .deleteOne( ) & as argument → use ($set.) func^n
- (b) Execute .deleteMany( ) & as argument → use ($set) func^n

3. What resources / materials / equipments / tools did you use for this activity ?

1. Online Resources:
   - MongoDB Community Server v4.4.5
2. (https://mongodb.com/try/download/community)
   - MongoDB Official Documentation
3. (https://www.mongodb.com/docs/atlas)
4. - MongoDB Compass Download.
   (https://mongodb.com/download-center/compass)
5. Equipments: PC Intel(R)_Core(TM) i5-6400@2.70Hz
   LENOVO_MT_30_AS_BU
   LENOVO_FM_ThinkStation_P310
   - Microsoft Windows 10 Pro 10.0.19044 Build
   - Screenshot → Screenshot application.
6.
7. Materials:
   - Lecture PPT for CRUD operations
8.

4. What skills did you acquire ?

1. W5HH abt UPDATE & DELETE operations.
   - How, to execute it in MongoDB?
2.
3. Implementation of queries on the MongoDB compass.
4. Algorithms behind the inbuilt function provided by Mongodb such as
   - UpdateOne()
   - UpdateMany()
5. (Query Evaluation Engine of MongoDB at the Runtime of server.)
6.
7. Reading the MongoDB ATLAS dasboard to understand the Cluster0 metrics.
8. Understanding How JSON format schema is used for argument)
   - RAM
   - No. of HITS
   - No. of MISS

5. Time taken to complete the activity ? 02:00 (hours)

Gayatri A

(To be filled by Faculty)

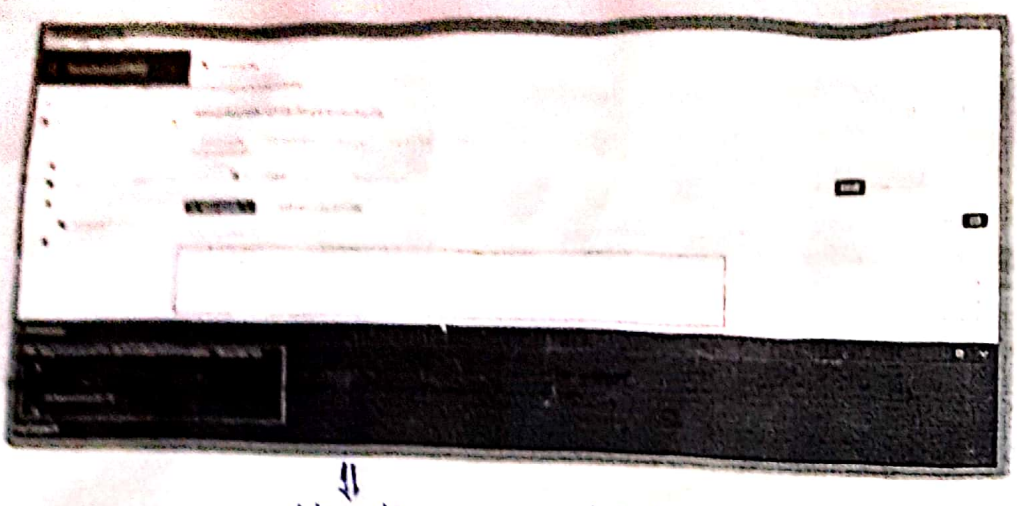
S. No.	Skills / Competencies	(Achieved / Not Achieved) (YES / NO)
01.	Performing CRUD operations (i.e. UPDATE & DELETE) operations on the MongoDB Community Server (V6.0)	
02.	Implementing and ~~Erase~~ Updation, Deletion of entries data in our database	
03.	Implementation of inbuilt-functions provided by MongoDB. i.e. [updateOne(), updateMany(), deleteOne(), deleteMany()]	
04.	Real World usecases of MongoDB	

Remarks

Total marks _____ out of 10.

Sign of Faculty

Date:



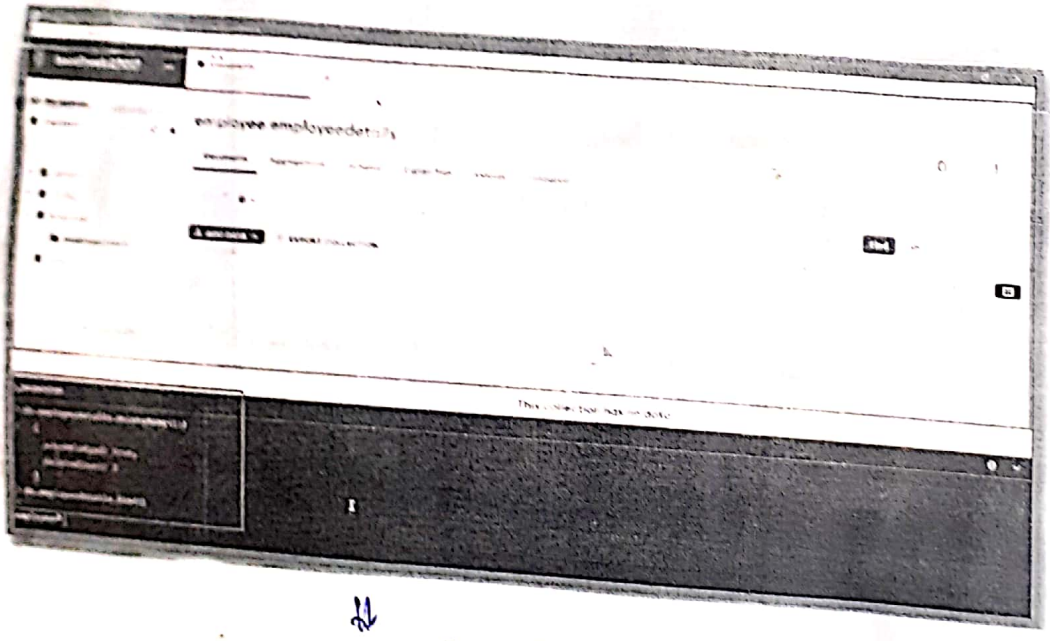
Step_03: In this screenshot, we are performing one of the CRUD operation i.e

DELETE

using inbuilt function of MongoDB database i.e.

.deleteOne( )

Now, Here we are deleting that datafield WHERE 'firstname': 'Saurav'



Step_04: In this screenshot, we are perfoming DELETE operation & DELETEing Many Records of the MongoDB database (i.e. employeedetails) at the same point of time.

.deleteMany({ })

Now, Here in this screenshot.

(i) we execute 'Mongoshell'

↳ then we Run '.deleteMany()' function.

(ii) Entire database {employee} & (employeedetails)

MONGODB CRUD Operations

(i) CREATE : Create or Insert operations add new documents to a collection. If the collection doesn't currently exist, insert operations will create the operation.

Eg:

```
db.employeedetails.insertOne(
{
  first name: "Sawan",
  last name: "Singh",
  batch: "B2"
}
)
```

MongoDB provides the following methods to insert documents into a collection:

- db.collection.insertOne()
- db.collection.insertMany()

In MongoDB, insert operations target a single collection. All write operations in MongoDB are atomic on a level of single document.

(ii) READ: Read operations retrieve documents from a collection, i.e. query a collection for documents.

Eg:

```
db.employeedetails.find();
```

MongoDB provides the following methods to read documents from a collection:

- db.collection.find()

You can specify query filters or criteria that identify that documents that return.

(iii) UPDATE: Update operations modify existing documents in a collection.

Eg:

```
db.employeedetails.updateOne()
db.employeedetails.updateMany(
  {
    firstname: "
    batch: {
  } ( $set: {batch: "B3"}
)
```

MongoDB provides the following methods to update documents of a collection:

- db.collection.updateOne()
- db.collection.updateMany()
- db.collection.replaceOne()

In MongoDB, update operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

(iv) DELETE: Delete Operations remove documents from a collection.

Eg:

```
db.employeedetails.deleteOne()
db.employee details.Many( )
  }
).{
```

MongoDB provides the following methods to delete documents of a collection :-

- db.collection.deleteOne()
- db.collection.deleteMany()

In MongoDB, delete operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

YASH GUPTA

CODE REPRESENTING MONGODB

Post.js

```
1  import mongoose from "mongoose";
2
3  const postSchema = mongoose.Schema(
4    {
5      userId: {
6        type: String,
7        required: true,
8      },
9      firstName: {
10       type: String,
11       required: true,
12     },
13     lastName: {
14       type: String,
15       required: true,
16     },
17
18
19
20
21     likes: {
22       type: Map,
23       of: Boolean,
24     },
```

→ FileName: Post.js

→ Code-Explanation:

↳ (a) Creating (FORM schema) for our project.

(b) Rows: ≡ Docs

Tuples:

userId, firstName, lastName, location, description, picturePath, userPicturePath

Post.js

```
14       type: String,
15       required: true,
16     },
17     location: String,
18     description: String,
19     picturePath: String,
20
21     likes: {
22       type: Map,
23       of: Boolean,
24     },
25     comments: {
26       type: Array,
27       default: [],
28     },
29   },
30   timestamps: true
31 }
32
33 const Post = mongoose.model("Post", postSchema);
34
35 export default Post;
36
```

→ File Name: Post.js

→ Code Explanation:

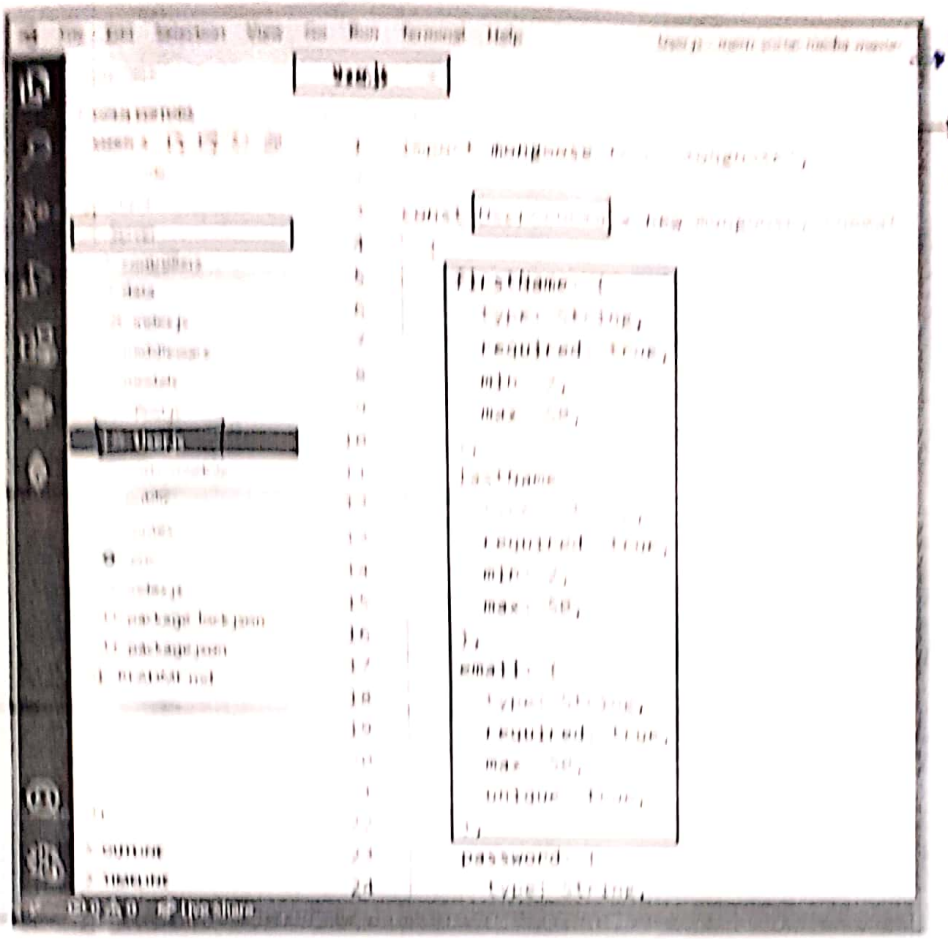
Here, we are storing various data-fields such as:

location:] String
description:]

comments ; likes

comments → type: Array

likes → type: Map



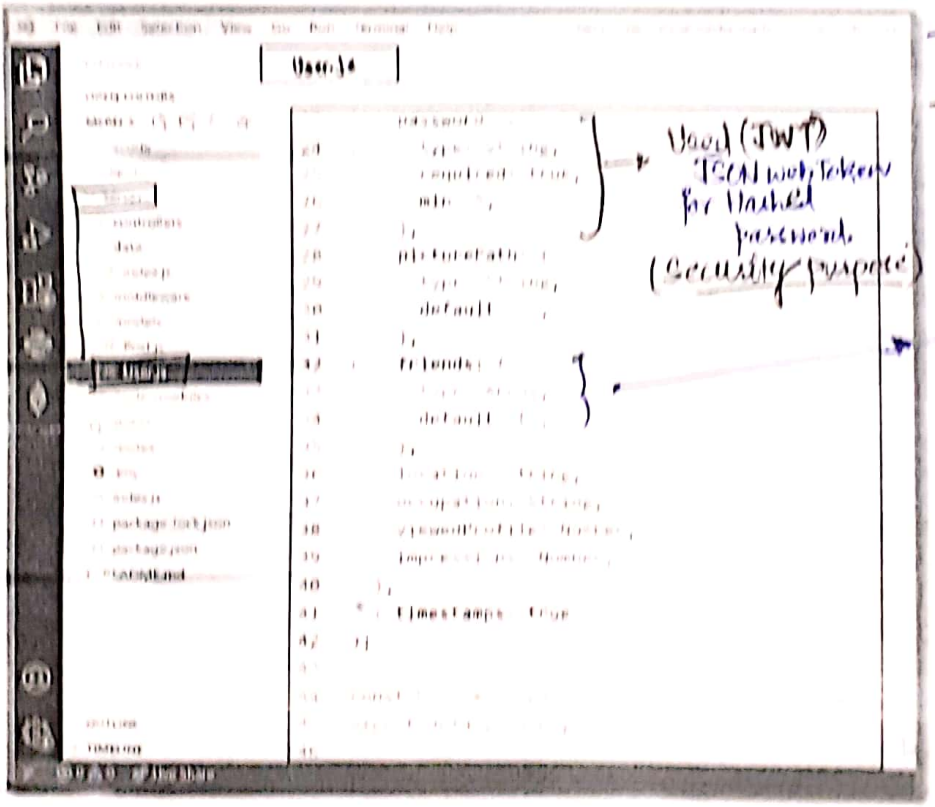
→ FileName: Users.js

→ Code Explanation:

↳ In this Users file, we are (authentication) creating a database.

↳ where we store firstName & lastName

- email
- password
- (picturePath)
- friends
- (timestamps)



→ FileName: Users.js

→ Code Explanation:

We are explaining

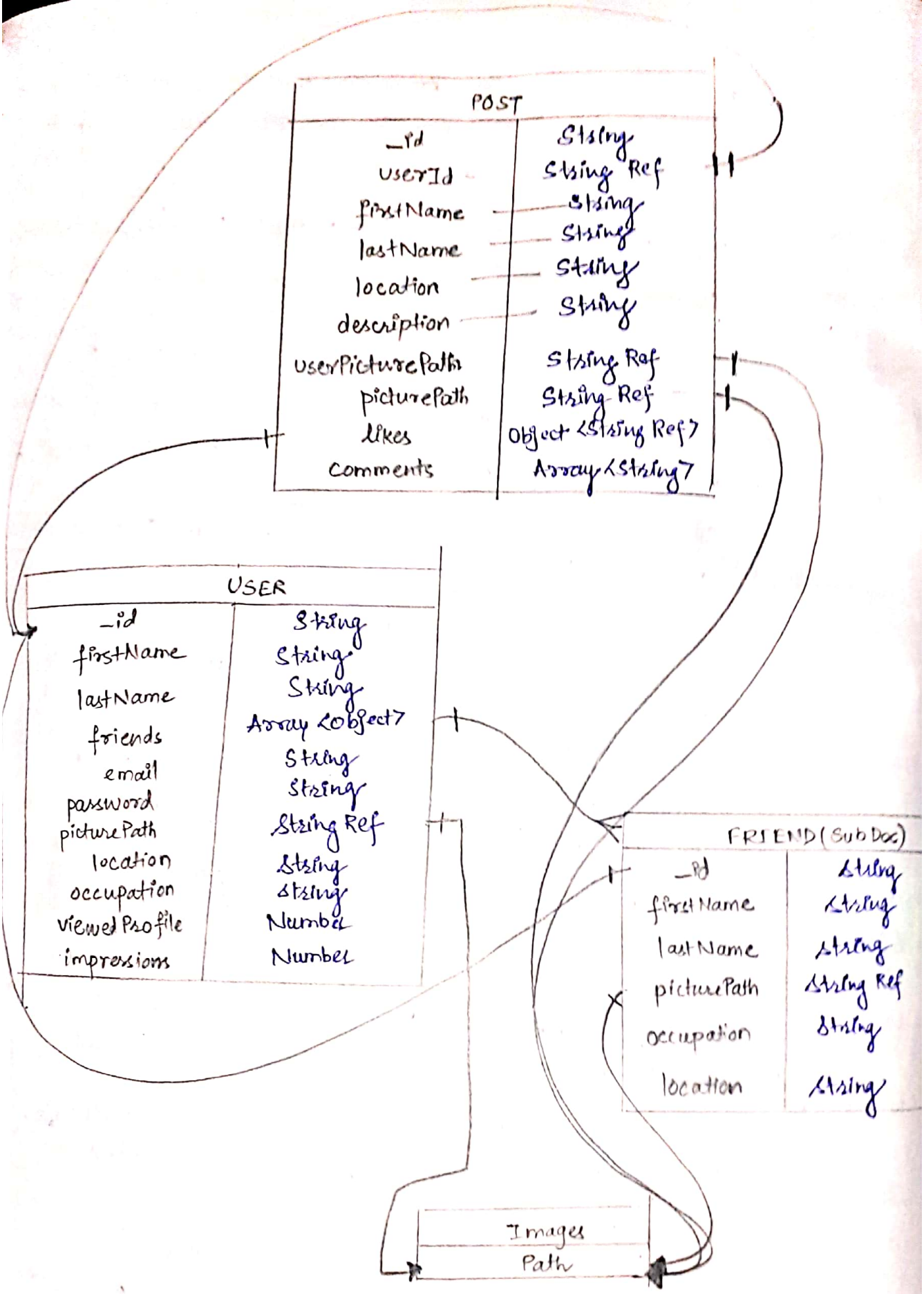
↓

"friends" feature

→ (Every Entities to Facebook, Twitter)

& the datatype is: "Array"

User (JWT) Json web tokens for Hashed password (Security purpose)



DATABASE SCHEMA ~~ER~~ DIAGRAM
UML

SKILL ACTIVITY NO: 4

Date 23/03/2022

(To be filled by the Student)

Title:

Designing a NoSQL Database for a Project

1. What is the purpose of this activity ? (Explain in 3 – 4 lines)

The purpose of this activity are as follows:-

(i) To understand the Need of database - [which we need → SQL databases, NoSQL databases]
- Tradeoff between the SQL and NoSQL database.

(ii) To understand in NoSQL databases → which one is best for our project use case. Finding the Properties of Document Based DB. Eg:

(iii) Now designing the database schema → what will the structure of our database.
- How many fields are required?
- What will be the minimum size(byte/bitsize) of various fields & also their datatype.

(iv) Future scope of our database schema → Optimization → Master-Slave Clusters Architecture → for better manageability of database.

2. Step performed in this activity (Explain in 5 – 6 lines)

Steps performed in this activity are as follows:-

Step (i): Firstly, understand all the requirements of database design i.e schema. Then design it and program it using Javascript.

Step (ii) :- Open VScode. Then create directory structure. → server → data → models → Post.js, User.js

Step (iii) → As, the database is using POST method., therefore, we will

Step (iv) → Now, we will use 'Mongoose' Mongo-DB (SDK library) & implement the schema using Javascript code.
- we are using
  - userId, dtype: string
  - firstName, dtype: string
  - lastName, dtype: string
  - likes, dtype: map

3. What resources / materials / equipments / tools did you use for this activity ?

1. Online Resources:
   - MongoDB Community Server
2. (https://mongodb.com/try/download/community)
   - MongoDB Official Documentation
3. (https://mongodb.com/docs/atlas)
   - MongoDB JS SDK download.
4. (https://mongodb.com/download-center/compass)

Dependencies:
react: '11.10.6',
style: '11.10.6',
'react-material': '5.11.16'
'redux-dom': '18.2.0'

Node: (v18.15.0)
Express: (v4.8.12)
→ All of them downloaded

5. Equipments: Laptop: NBQBRJST, Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80GHz
   - 4.00 GB (3.98 GB) Memory
   - VS Code Editor
7. Materials:
   - github.com/YG3AFD
   - db.diagrams.net

4. What skills did you acquire ?

1. W5H1H about (ER) Diagram & (UML) diagram
2.
3. Realworld implementation & designing of Database Schema in UML website.
4.
5. What, Why & Where for Data Modelling
6. Understanding about NoSQL database schema design
7. Difference btw NoSQL document based schema & SQL Based Schema (design)
8. Defining, Planning of Datatypes of columns in MongoDB

5. Time taken to complete the activity ? 05:00 (hours)

Signature of Student

(To be filled by Faculty)

S. No.	Skills / Competencies	(Achieved / Not Achieved) (YES / NO)
01.	Designing a NoSQL database using MongoDB -Atlas. (CloudVersion of MongoDB)	
02.	Data Modelling and ER Diagrams ↳ Handmade sheet at I page of SK-04.	
03.	React Redux file folder architecture Understanding & Implementation.	
04.	Understanding of Redux Concept & Implementation.	
05.	Understanding of Frontend User Interface. & implementation.)	

Remarks

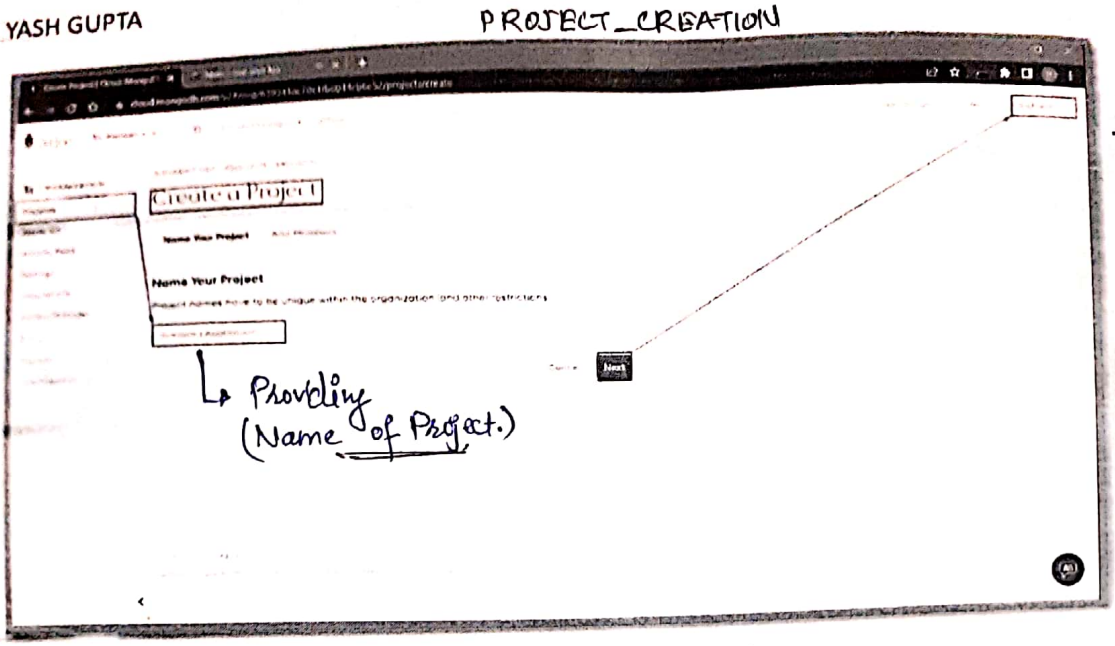
Total marks _____ out of 10.

Sign of Faculty

Date:

PROJECT_CREATION

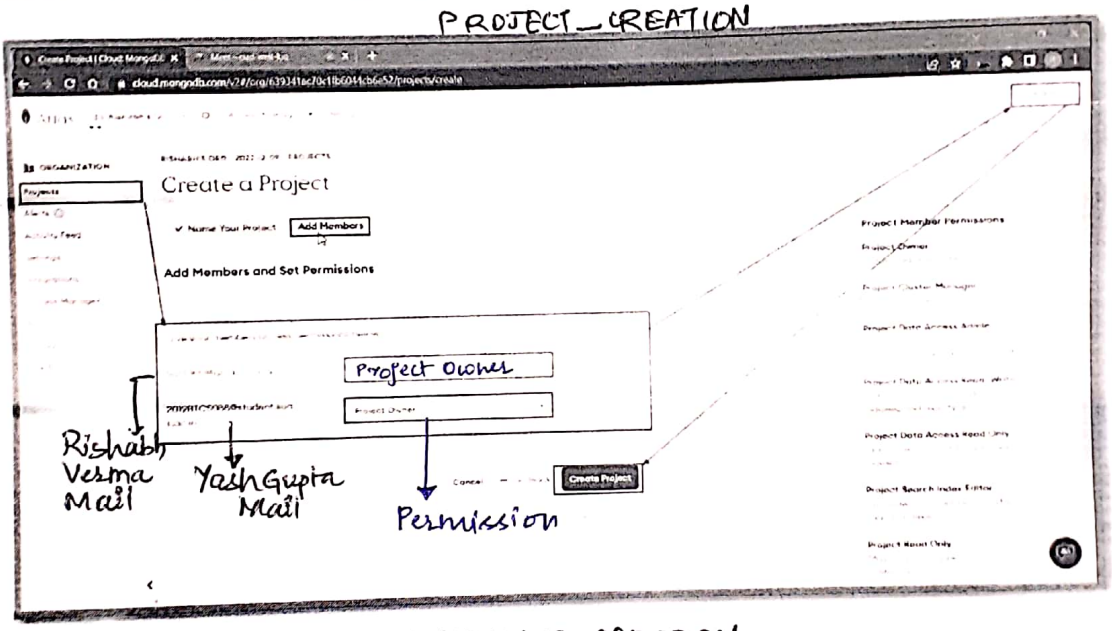
(1)



→ Create Project
Here, we are using (MongoDB ATLAS)
↓
cloud version where our Cluster0

PROJECT_CREATION

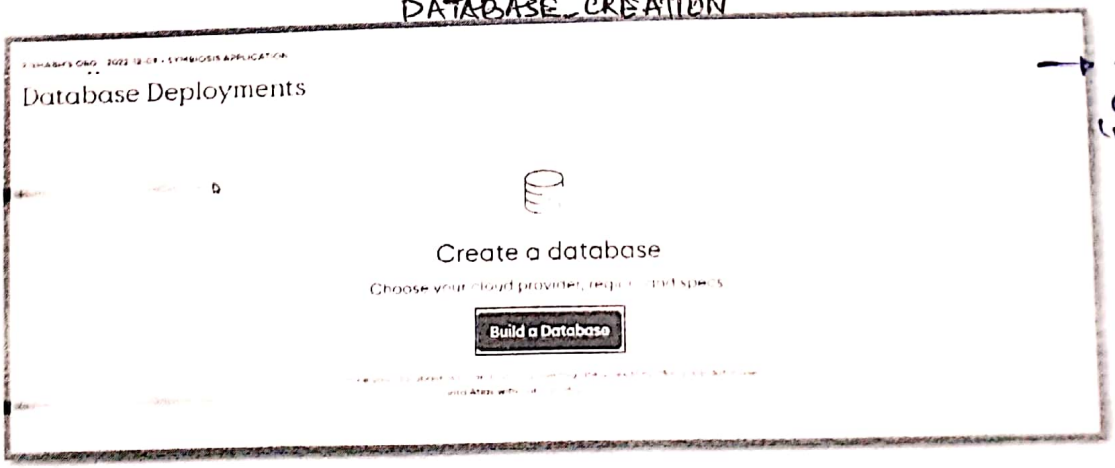
(2)



⇒ Adding the Members
↓
And providing the Necessary Condition (Permissions)

DATABASE_CREATION

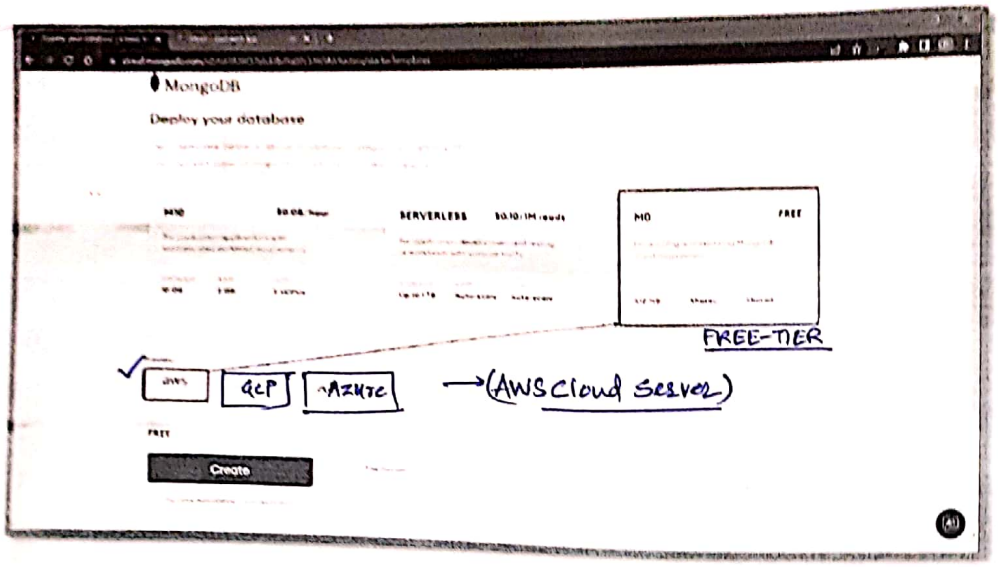
(3)



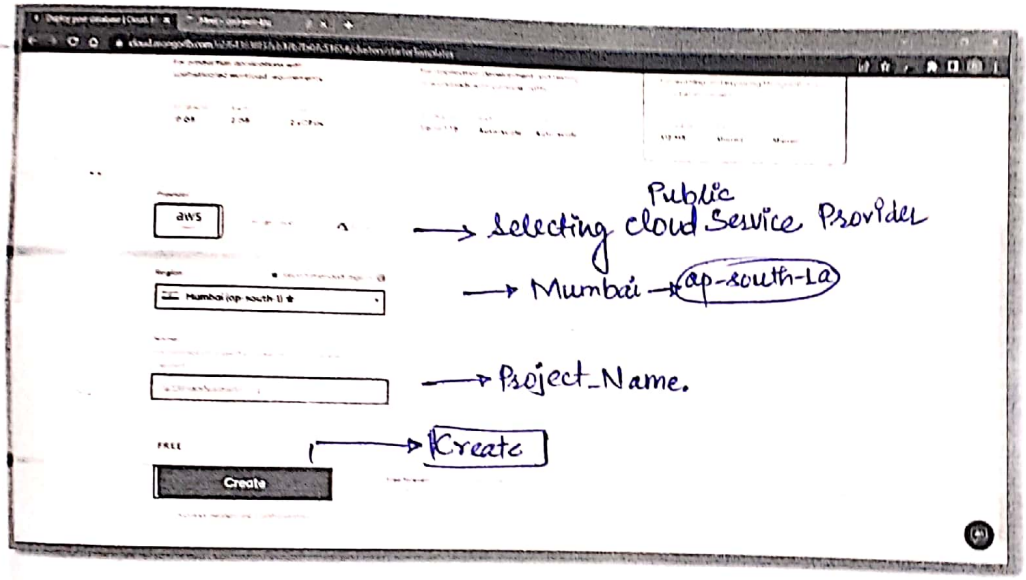
→ Simply, click at 'Build a Database'

For deployment of database server
↳ We are using FREE-TIER

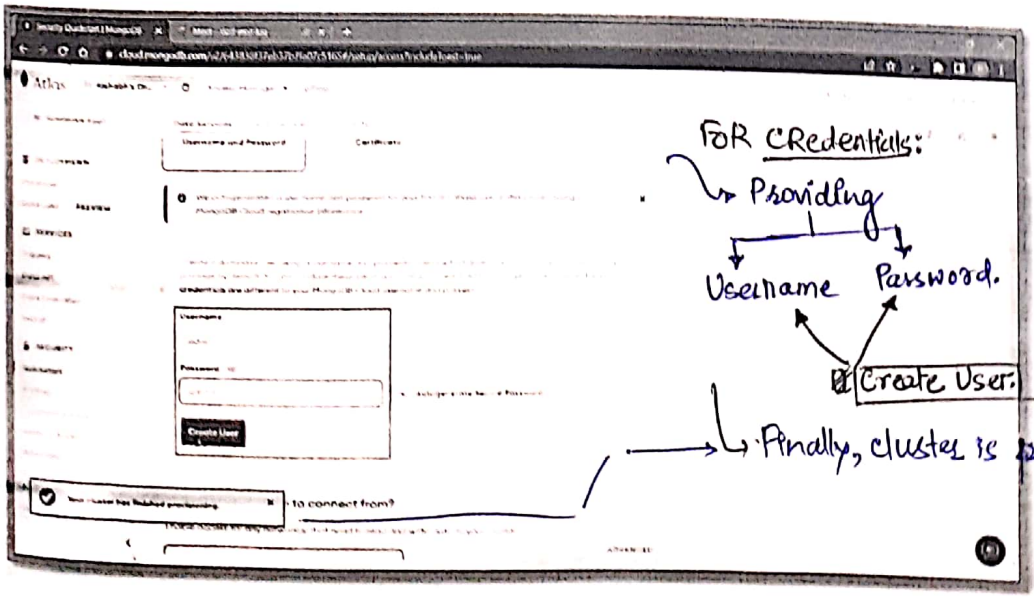
④

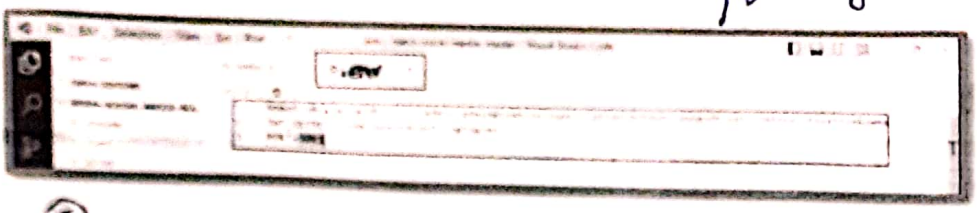


⑤



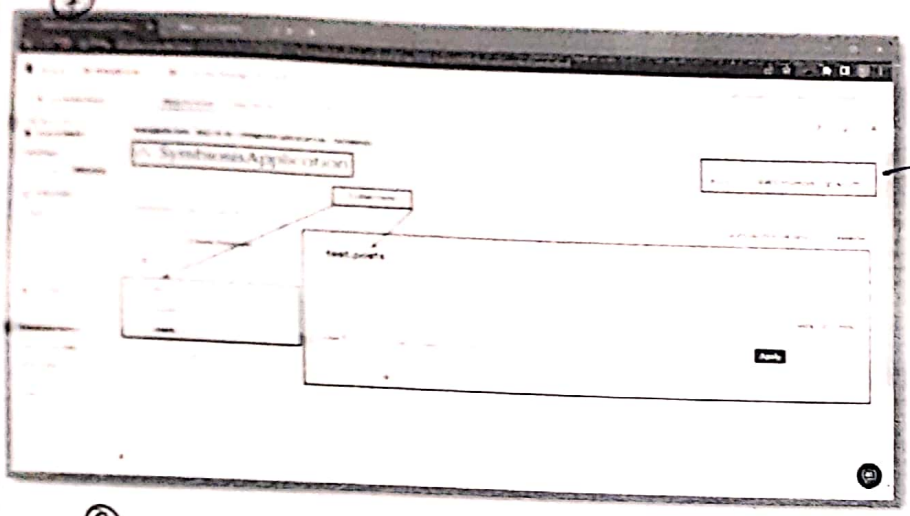
⑥





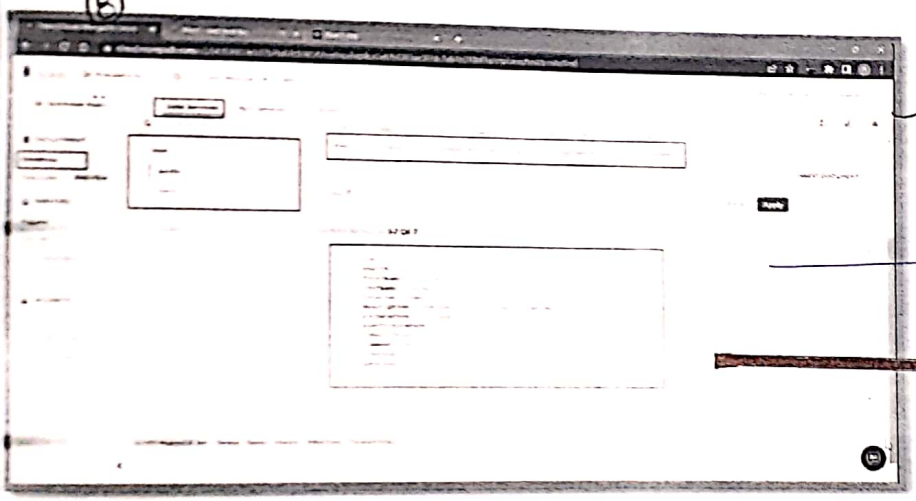
→ Setting up the → URL where our Cluster is deployed
→ Also, adding Username & password for Authentication.
→ Port: On which server is Running.

(7)



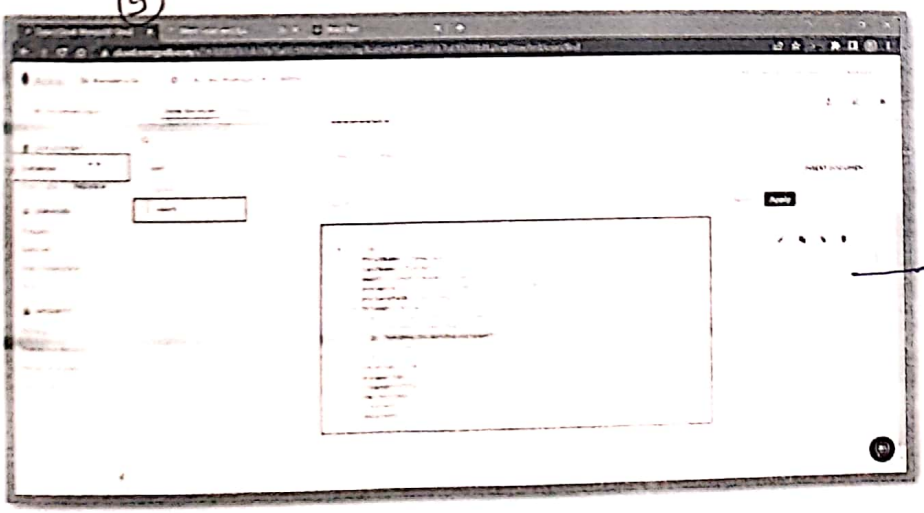
→ AWS_Cloud Region (Mumbai ap-south-1a) where our ab server is deployed.

(8)



→ All the data which sent by UserForm in Experimental Data is inserted in order to check about CRUD operations.
→ "POST.JS" page → "Post" in our

(9)

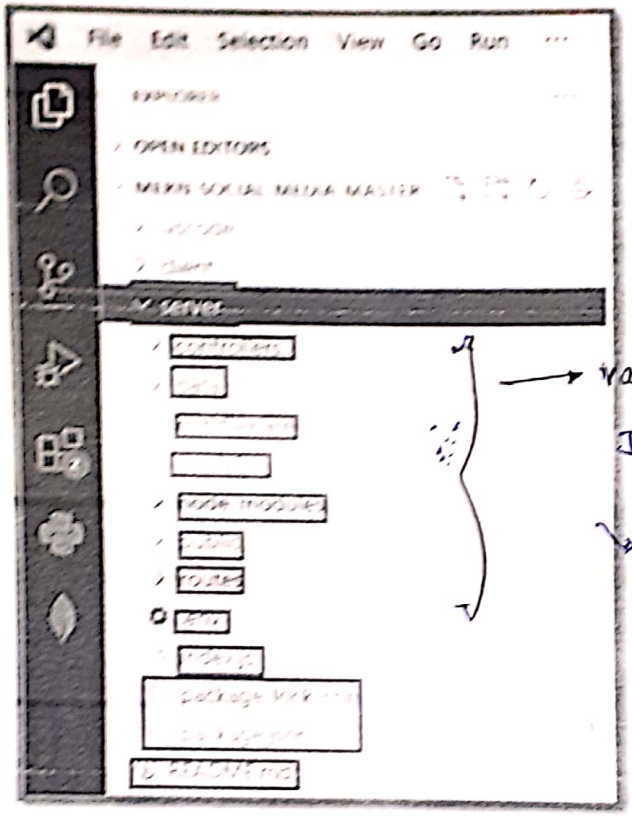


→ All the data which is credential Based
→ Email
Password –

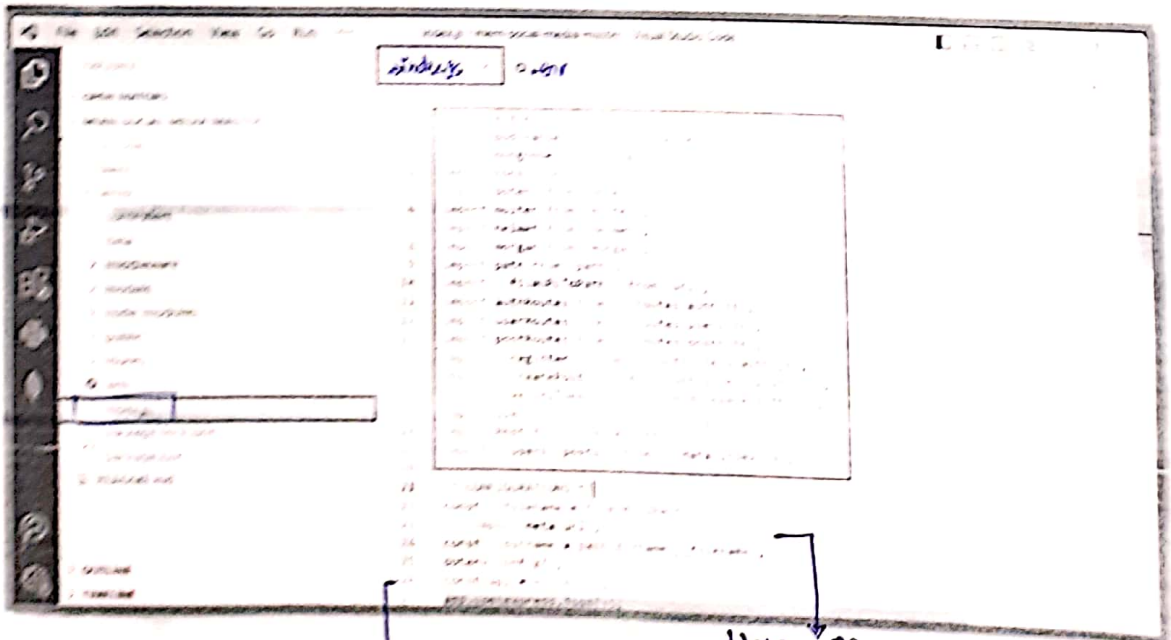
YASH GUPTA

SKILL 05

Codebase Server Side



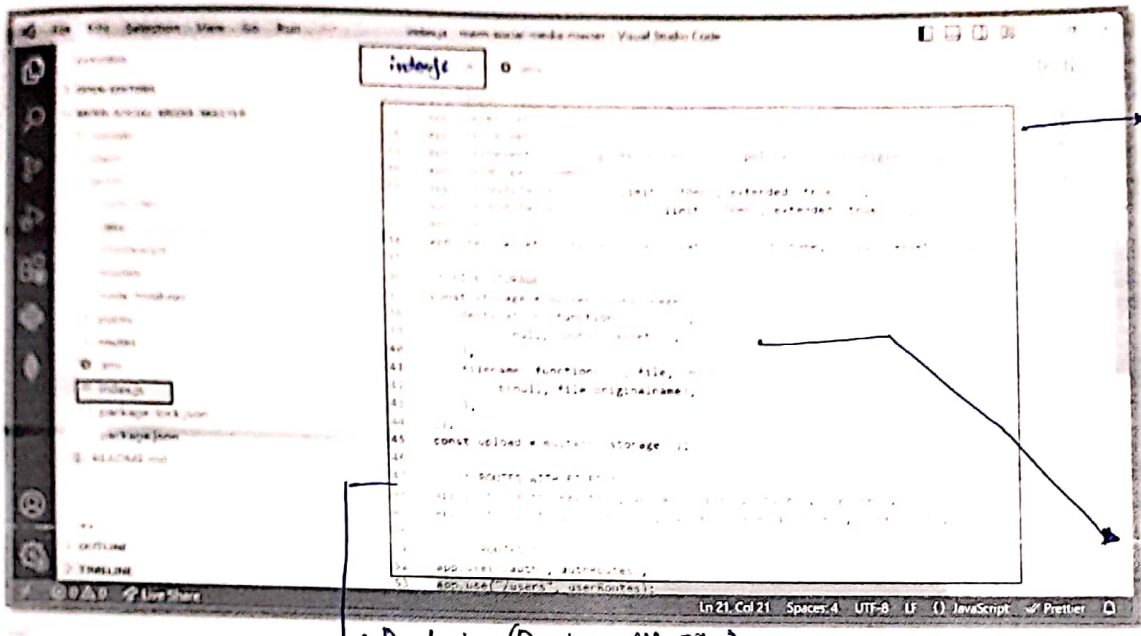
→ various directories created as per JS coding standards in order to manage.



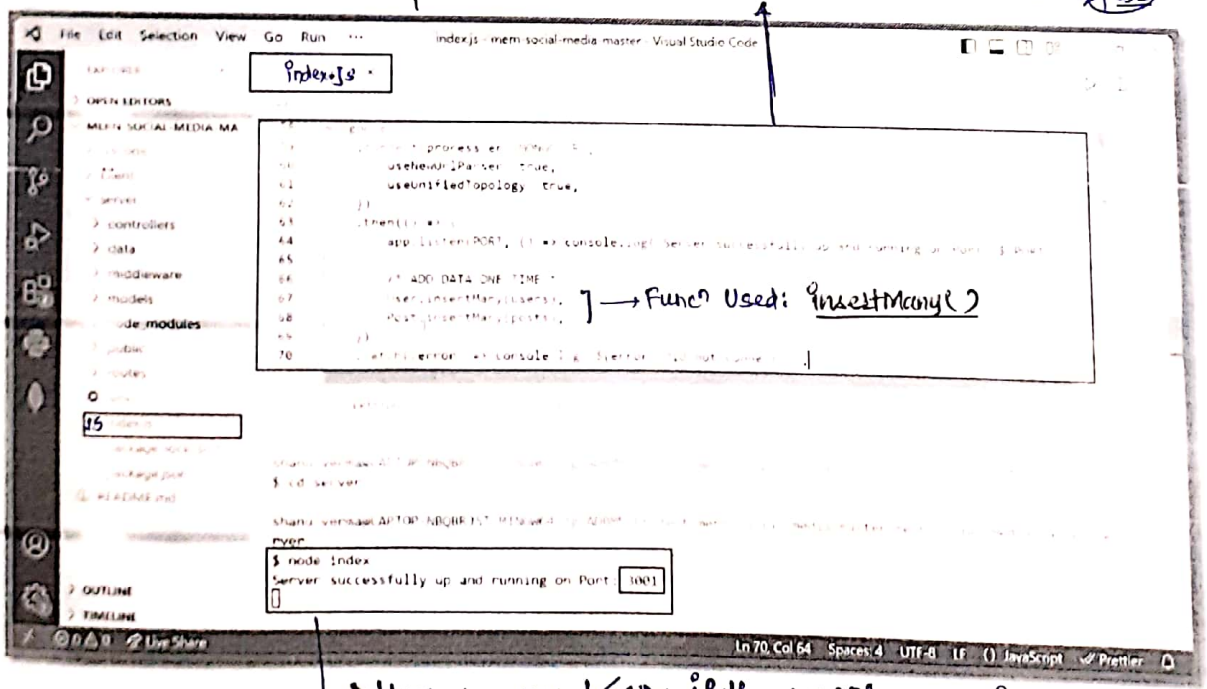
Line 26: 'app' is started via 'Express' server

Here, filename & directoryname are created.

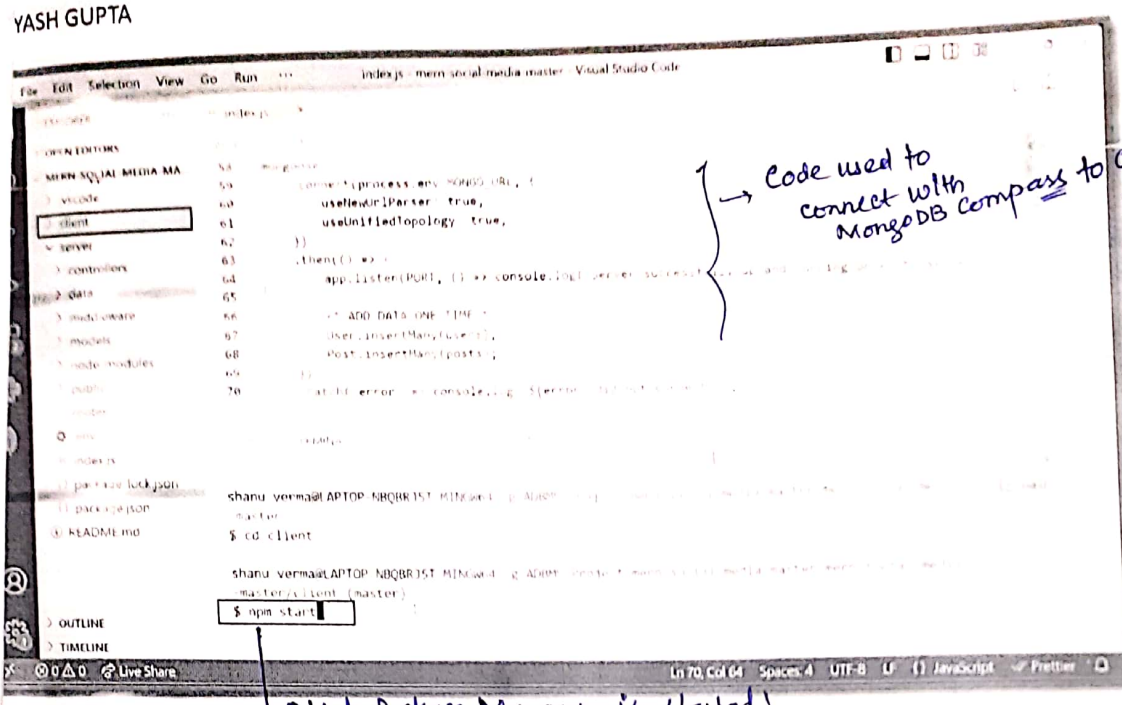
2019BTCS088



Code explanation: Here, the MongoDB → Fun → To connect with it, we are inserting Many → (users, posts)



Here, we are checking if the NODEjs server in port 3001 is Running or NOT?



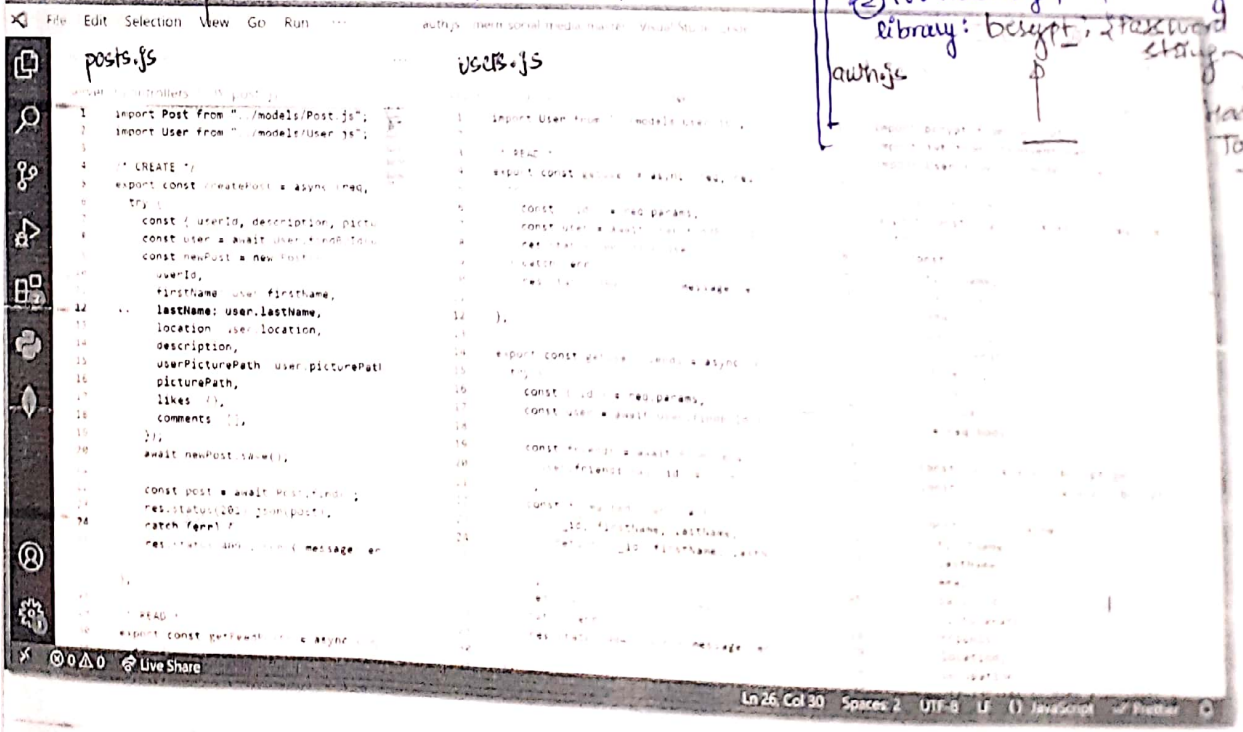
→ Code used to connect with MongoDB compass to community server.

→ Node Package Manager is started!

→ code for the Tweet/comment Posted by User in His/Her profile

(UI/UX)
→ Look & Feel of the User's profile is coded.

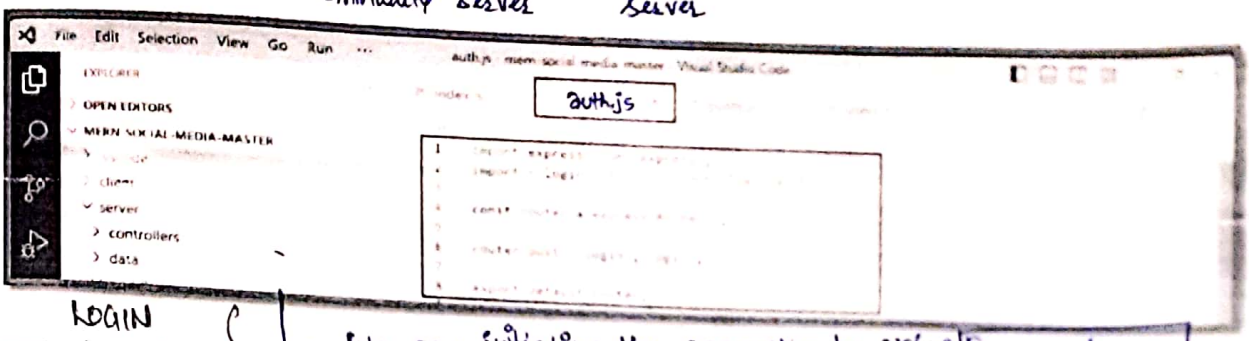
For authentication
↳① Creating JWT (Java Web Token)
②For Hashing purpose, using library: bcrypt; (Password string → Hashed Token)
auth.js



YASH GUPTA

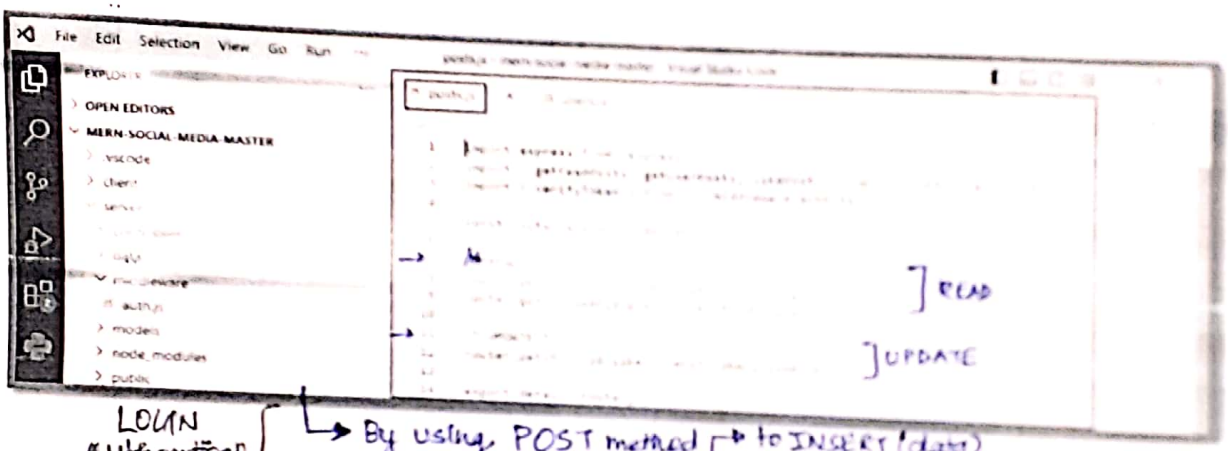
AUTHENTICATION PAGES in (Javascript)

- Community
- MongoDB server
- Express server



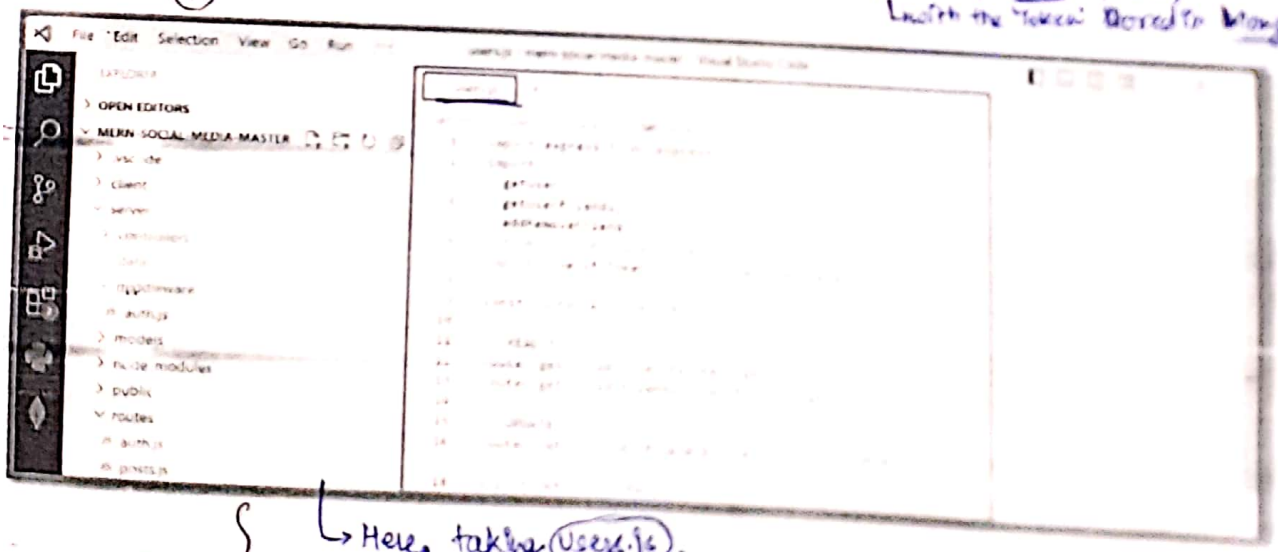
LOGIN Authentication PROCESS (i)

→ We are initiating the connection by using Express server with login page



LOGIN Authentication Process (ii)

→ By using POST method
- to INSERT (data) & take it as input from user
- And verification of "token" {JWT} with the "token" stored in MongoDB



→ Here, taking (users.js),
- GetUser
- GetUserFriends

} → These two methods are called for displaying the details.

SKILL ACTIVITY NO: 5

Date: 19/04/2023

(To be filled by the Student)

Title: Implementation of a Database Management Project using MongoDB as Backend.

1. What is the purpose of this activity ? (Explain in 3 – 4 lines)

The purpose of this activity are as follows :-

(i) To understand about various types of Software architectures & which one fulfills the requirement in our Project.
- Master-Slave
- Client-Server
- Node-Agent

(ii) To understand how to deploy our Database Server, on which Public Cloud Service Provider, which region to deploy. Also, to ensure Maximum Security, as data is MOST CRUCIAL.

(iii) To understand better on Storage of Data
- which type of storage to refer? (Object Block File)
- How to implement CORS for better prediction of our datapoints, in order to improve business problem.

↳ Therefore, we deploy MongoDB in Azure cloud.

(iv) For Authentication & Authorization:
↳ Using Email, Password → How to understand best security principles acc. to our need & cost trade-off. we used which one tool / library?

2. Step performed in this activity (Explain in 5 – 6 lines)

Steps performed in this activity are as follows:-

SERVER SIDE

Step(i) For Server side, we need to install
- NodeJS : v18.10.
- Express Server :
- MongoDB Community Server :
- Postman

Step(ii) Then start creating the data modelling & ERDiagrams which provides flow of database schema.

Step(iii) Created credentials of NodeJS server for authentication.

Step(iv) Adding Backend data & testing Demo (if its working properly)

CLIENT-SIDE

Step(i): For Client, we need to install
- React library
- React Router
- Redux Toolkit

Step(ii) Using the credentials provided by NodeJS server & under controllers (auth.js) inserting the logical Business logic.

Step(iii) Setting up Routes → User Route, Post Route

Step(iv) Setting React Redux & React Router. → File architecture

Step(v) For Creating various components of Dashboard page.
- Color Theme
- Dark Mode
- Homepage & Widgets.
- Navbar
- Styling Setup
- Register, Login Pages & Form.

3 What resources / materials / equipments / tools did you use for this activity ?

1. Online Resources
2. MongoDB Community Server (https://www.mongodb.com/try/download/community)
3. MongoDB Official Doc (https://www.mongodb.com/docs/atlas)
4. MongoDB JS SDK Download (https://www.mongodb.com/downloadcenter/compass)
5. Equipments: Laptop: NBQBRJST, Intel (R) Core(TM)
   - i5-8265U CPU @ 1.60GHz
   - VS Code Editor
6.
7. Materials: [github.com/YG3A5D]
8.

Dependencies:
"mongodb": "^6.6.0"
"mongoose": "^8.4.4"
"cors": "^2.8.5"
"dotenv": "^16.4.5"
Nodemon (v3.1.0)
Express: (v4.19.2)

4 What skills did you acquire ?

1. Understanding of Client-Server Architecture. → How it works & How to implement?
2.
3. WSHH about Node, Express, MongoDB
4. Testing: Understanding about STLC & implementing Unit testing in our Project. (Software Testing Life Cycle)
5. Middleware architecture implementation using Node.js & Express.JS
6. For Authentication & authorization → Use of bcrypt & JWT (JsonWebToken)
7. Database: CORS application.
   Server → Launched own db cluster on AWS cloud Mumbai ap-south-1 region in order to implement "CORS"
8.

5 Time taken to complete the activity ? 202:00 (hours)

Signature of Student

(To be filled by Faculty)

S. No.	Skills / Competencies	(Achieved / Not Achieved) (YES / NO)
01.	Installation of Nodejs Server & deploying it on local PC at port 3001.	
02.	Installing multiple dependencies required in project for frontend & Backend both i.e. React, React-dom, etc.	
03.	Using "bcrypt" and "JWT" for Hashing our Users password & following basic Standards of Authorization & Authentication.	
04.	Understanding of CORS (Cross Origin Resource Sharing) ↳ used in AWS S3 Bucket.	
05.	Unit Testing performed for all the react components in [Front-Side Web Page.] (React Interfaces)	

Remarks

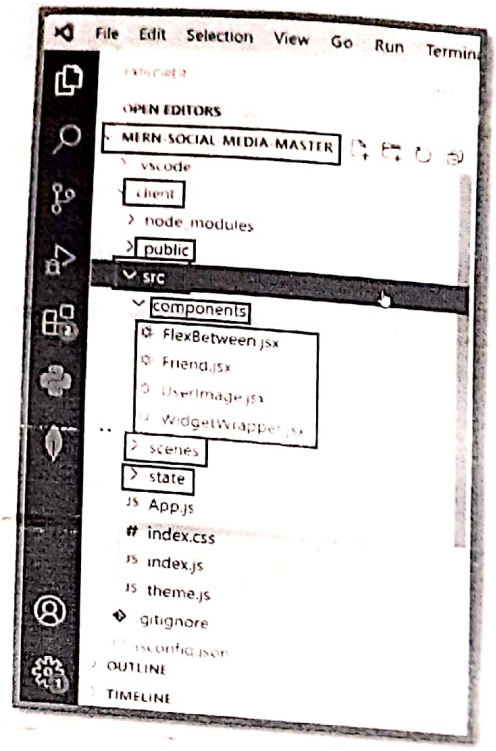
Total marks _____ out of 10.

Sign of Faculty

Date:

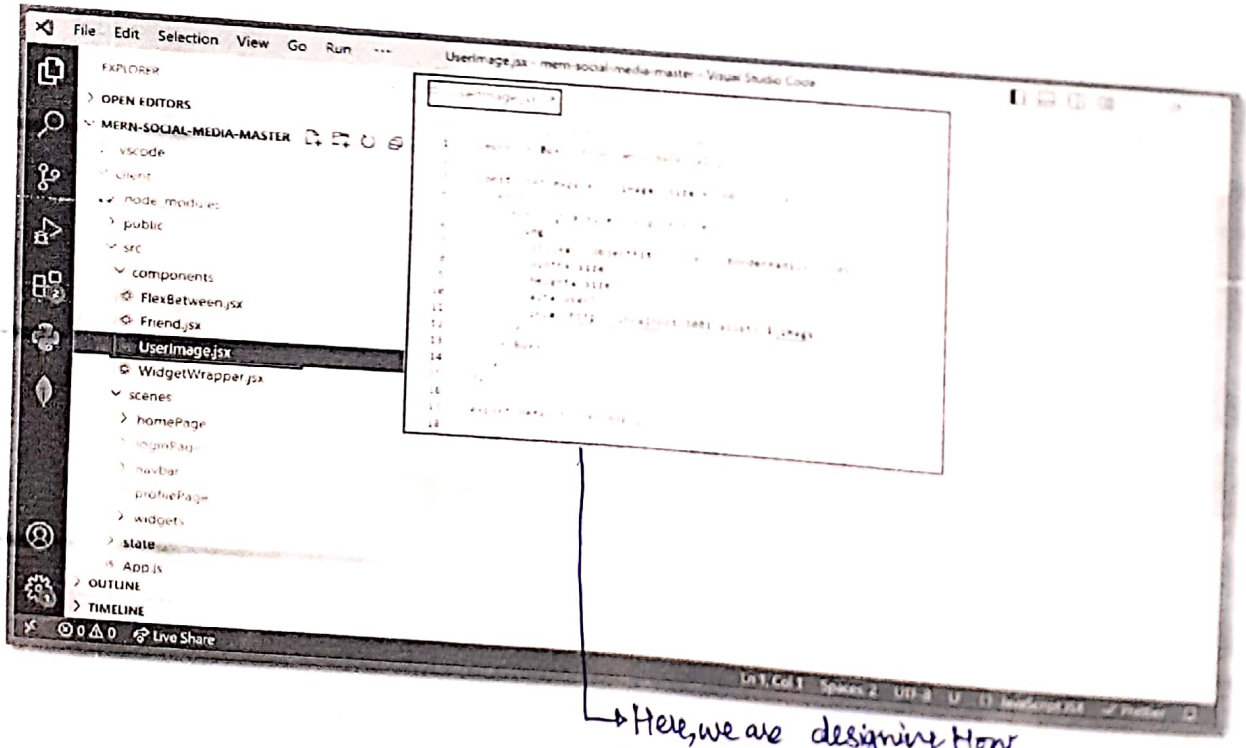
YASH GUPTA

Codebase: Client-Side



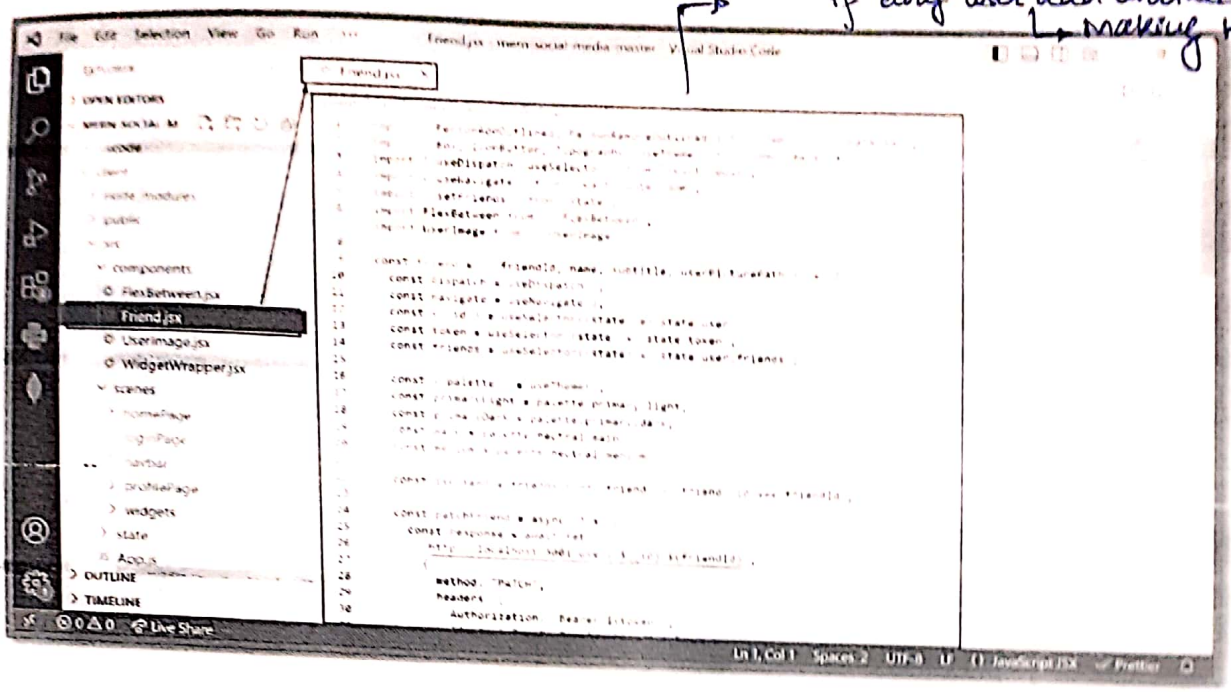
→ Screenshot showing ('client') directory Tree
↳ Installed all dependencies & their version under node_modules directory.

↳ Under 'src' directory,
↳ Created React-Components used in our project
→ FlexBetween.jsx
→ Friend.jsx
→ UserImage.jsx
→ WidgetWrapper.jsx



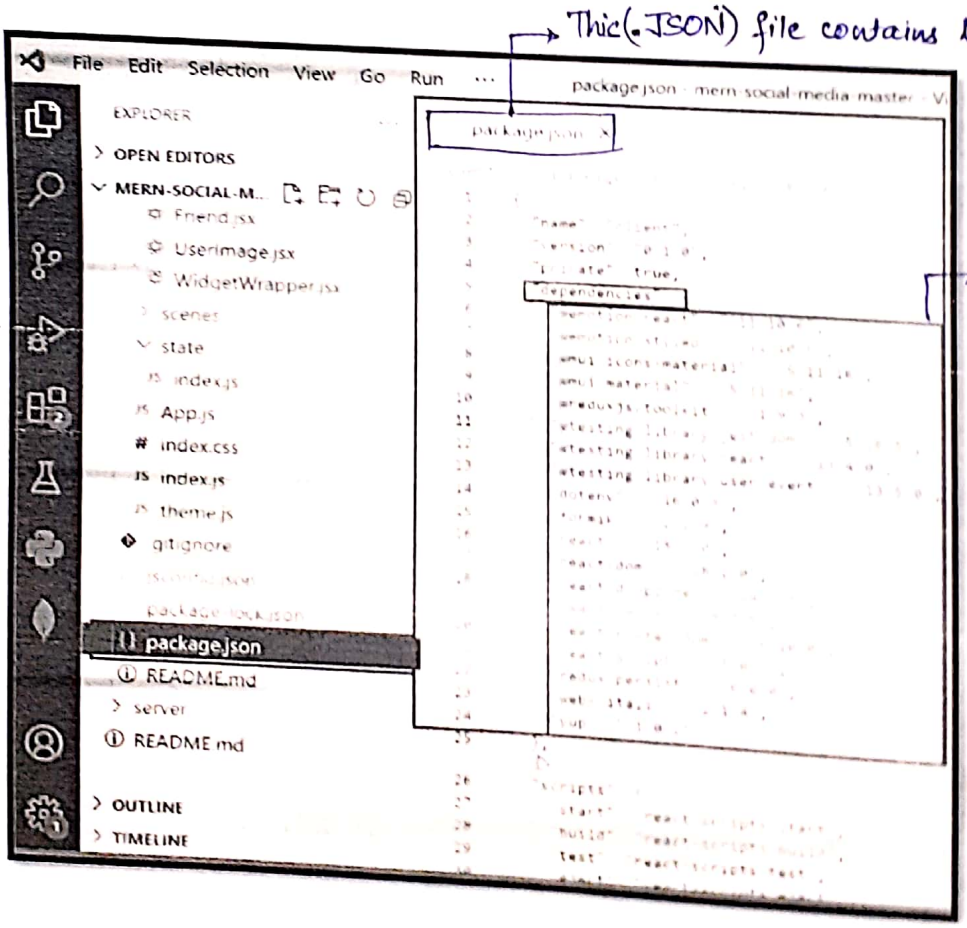
↳ Here, we are designing How User's Image will display by defining the 'width' & 'height'.

Here, we created the code logic if any user add another user making them/her friends



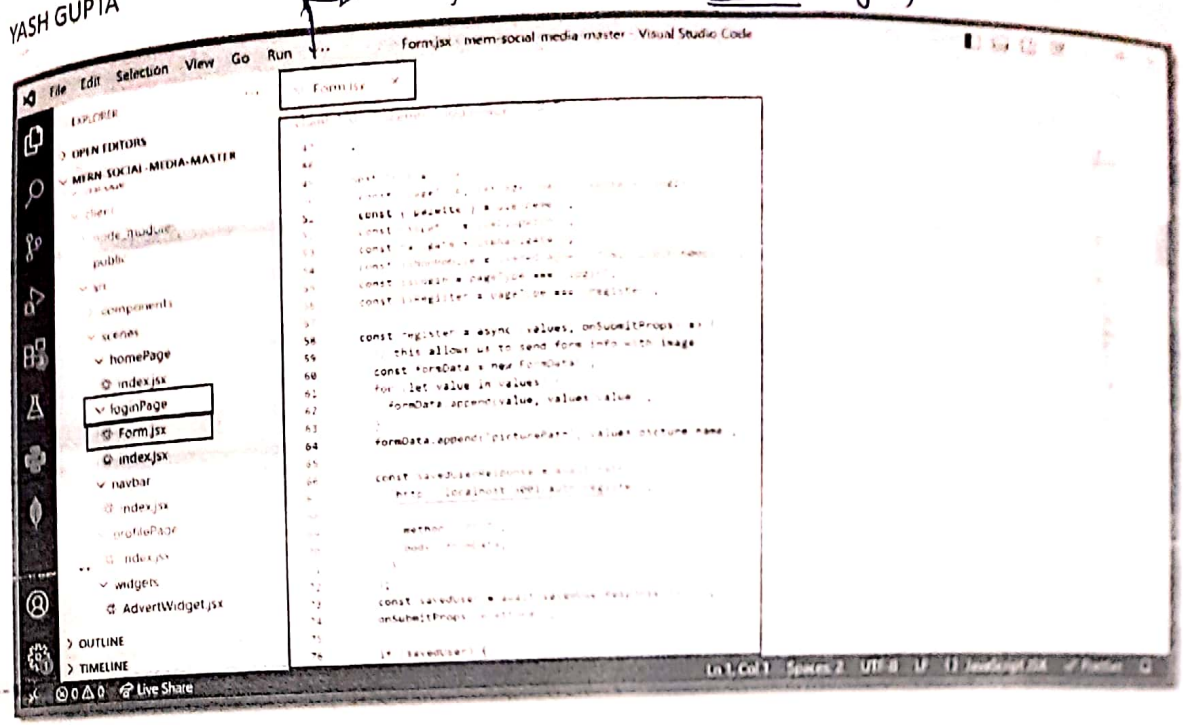
This (.JSON) file contains list of all dependencies used in our project.

These are the dependencies with their version.

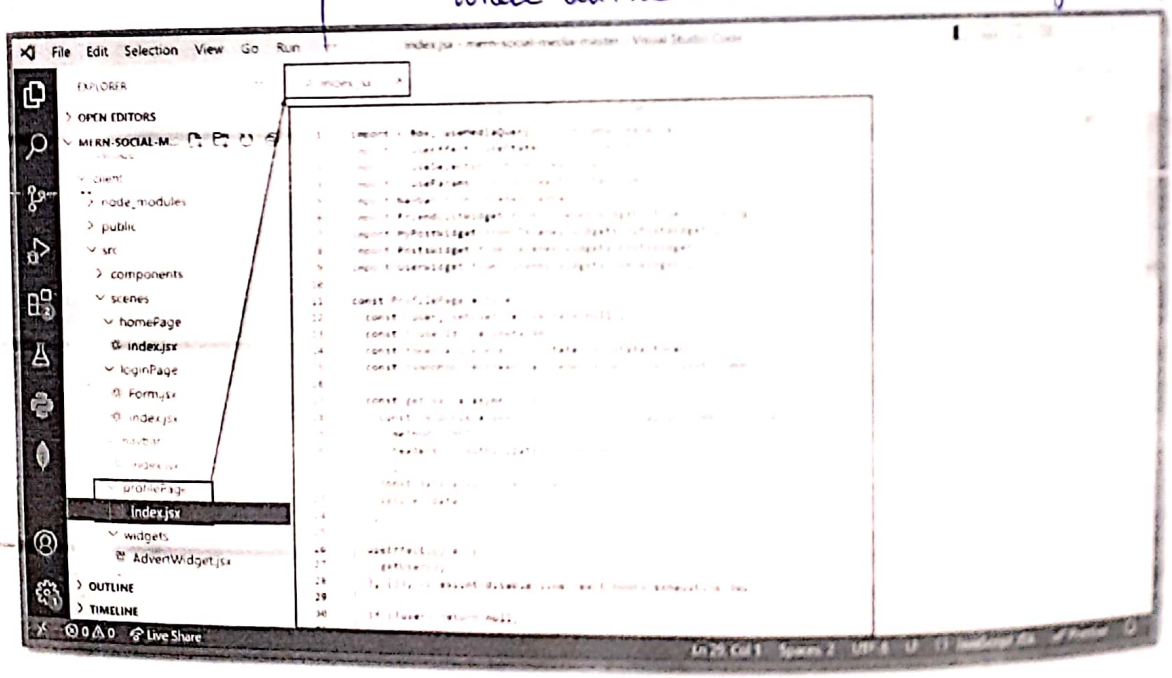


YASH GUPTA

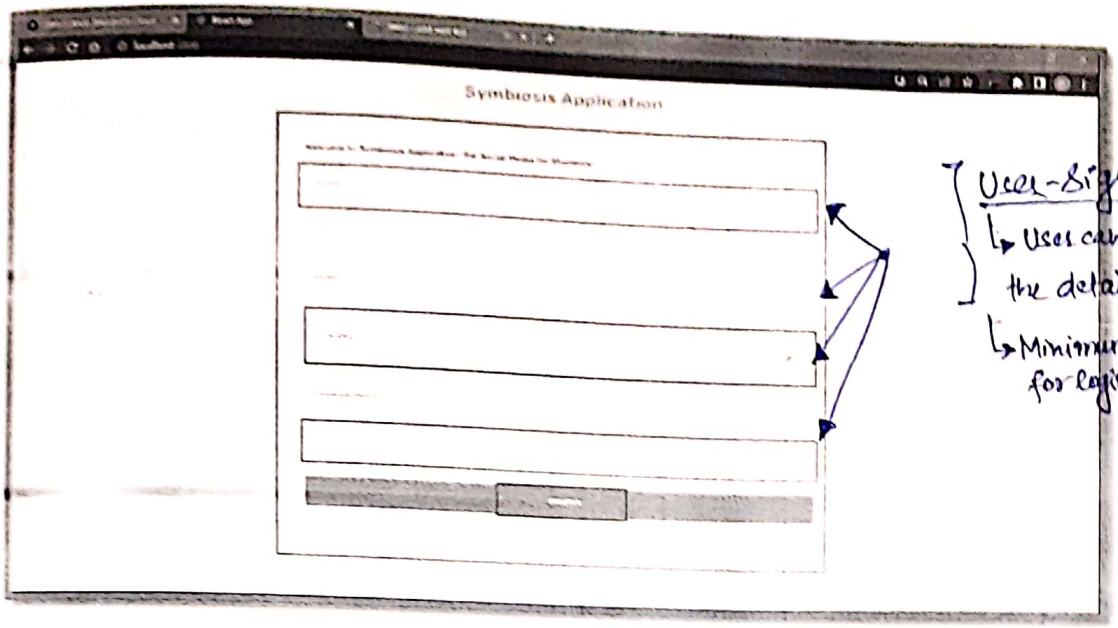
This file contains the LOGIN Page form code.



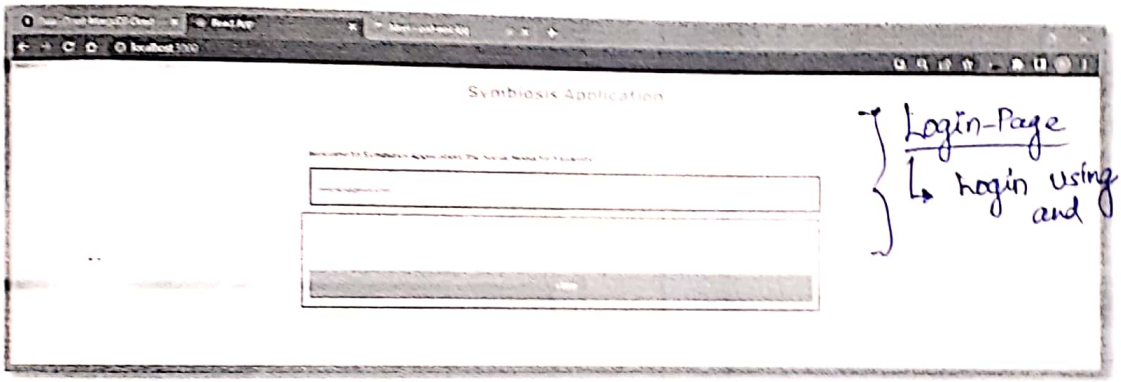
This file contains code of the DASHBOARD page.
Where all the activities related to User gets displayed.



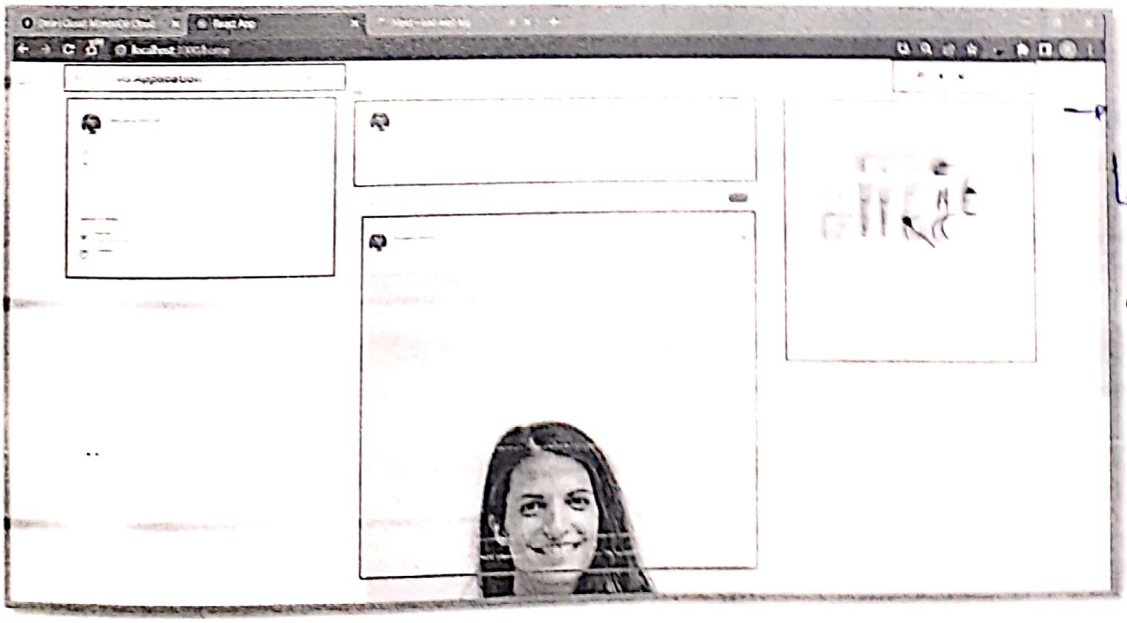
OUTPUT of Symbiosis_Application



User-SignUpPage
- Users can fillup all the details.
- Minimum Requirement for Login.



Login-Page
- Login using (Email-Address and Password.)



User Dashboard Page
- Show all the activities Post, Liked, Disliked, Comment, Trending among all other Users. in one page.