ASE_Lab7 Report

Implement Customer CRUD using MEAN Stack

Mean Stack Development

March 16, 2019

Documented by:

Dharani Muli (Class ID : 18),

Chakra Pavan Kumar (Class ID: 13)

Introduction

Objectives

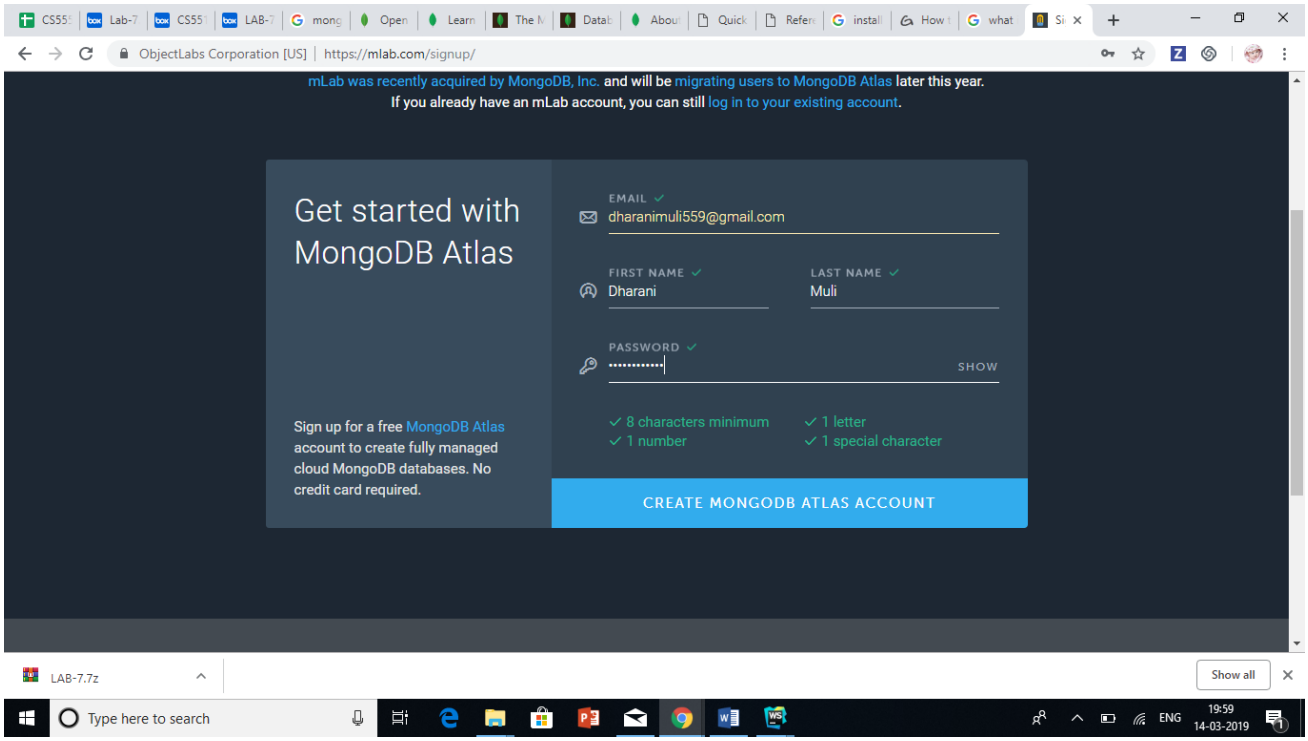
Develop customer CRUD API using MEAN Stack

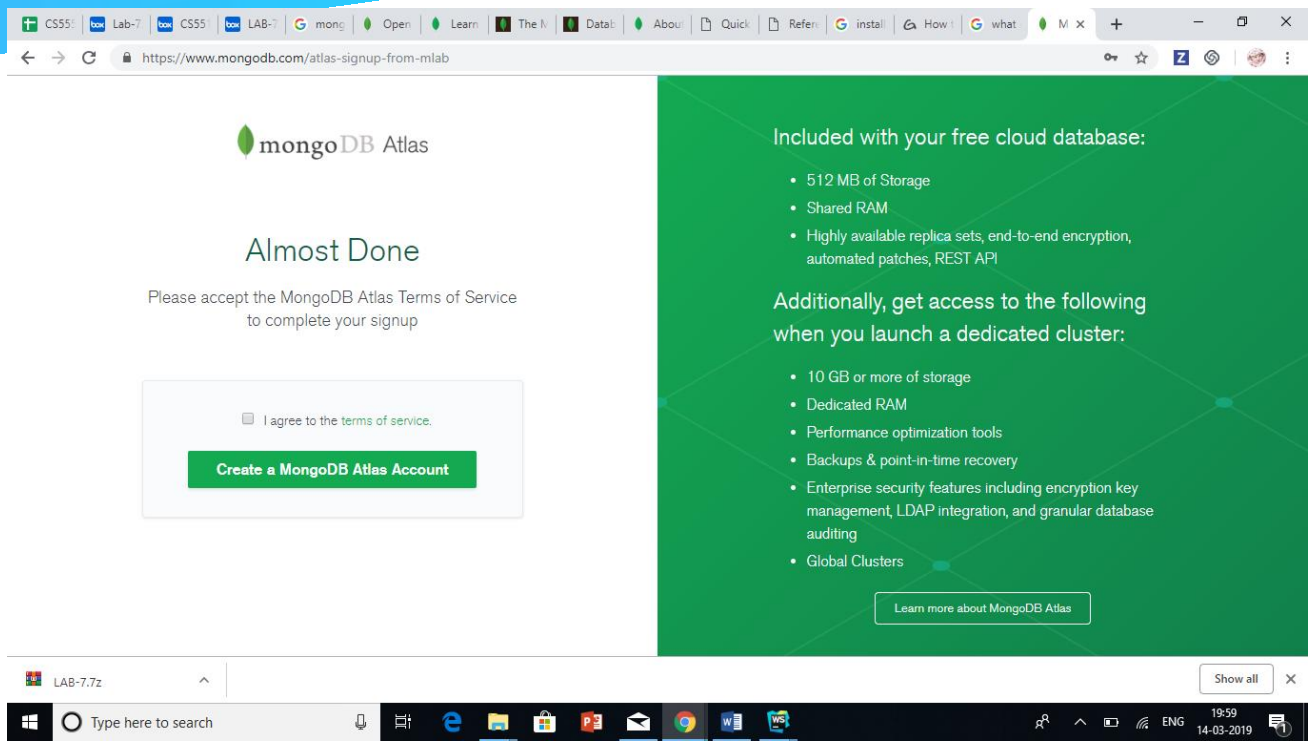
Design/Implementation

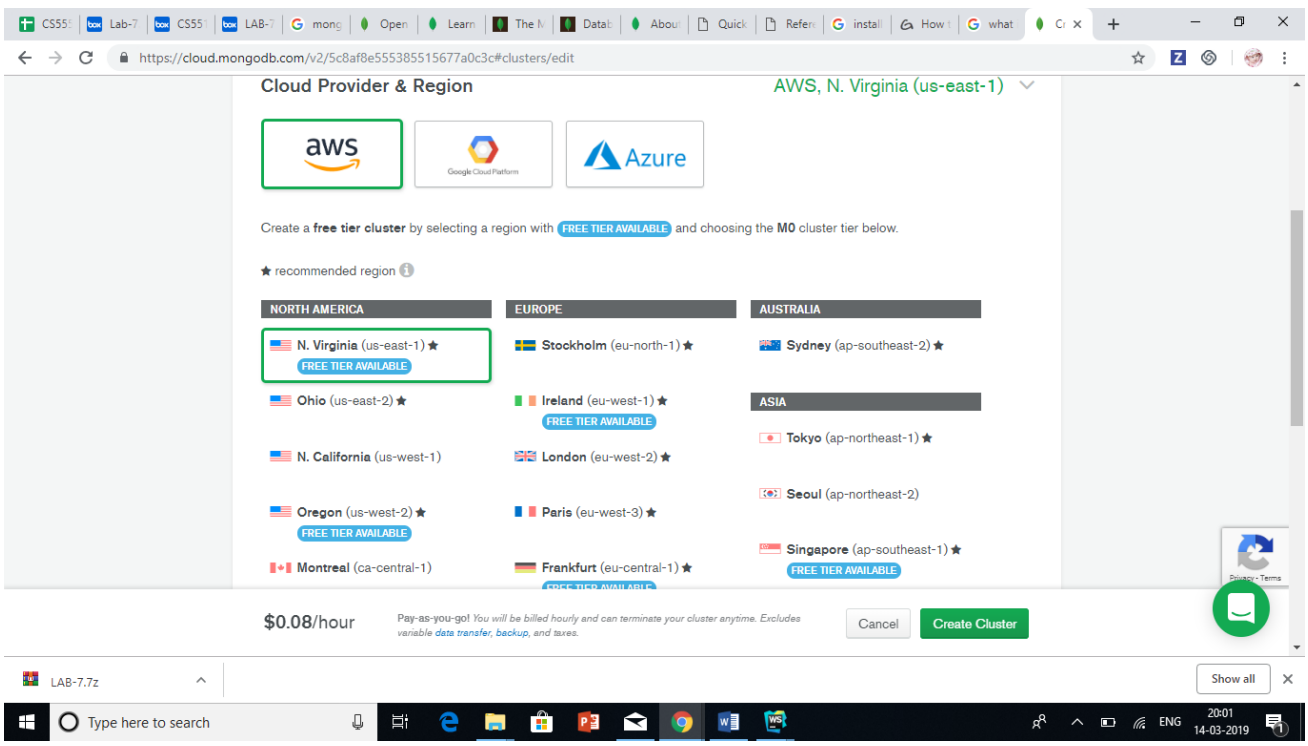
We have followed below steps to successfully complete this lab assignment:

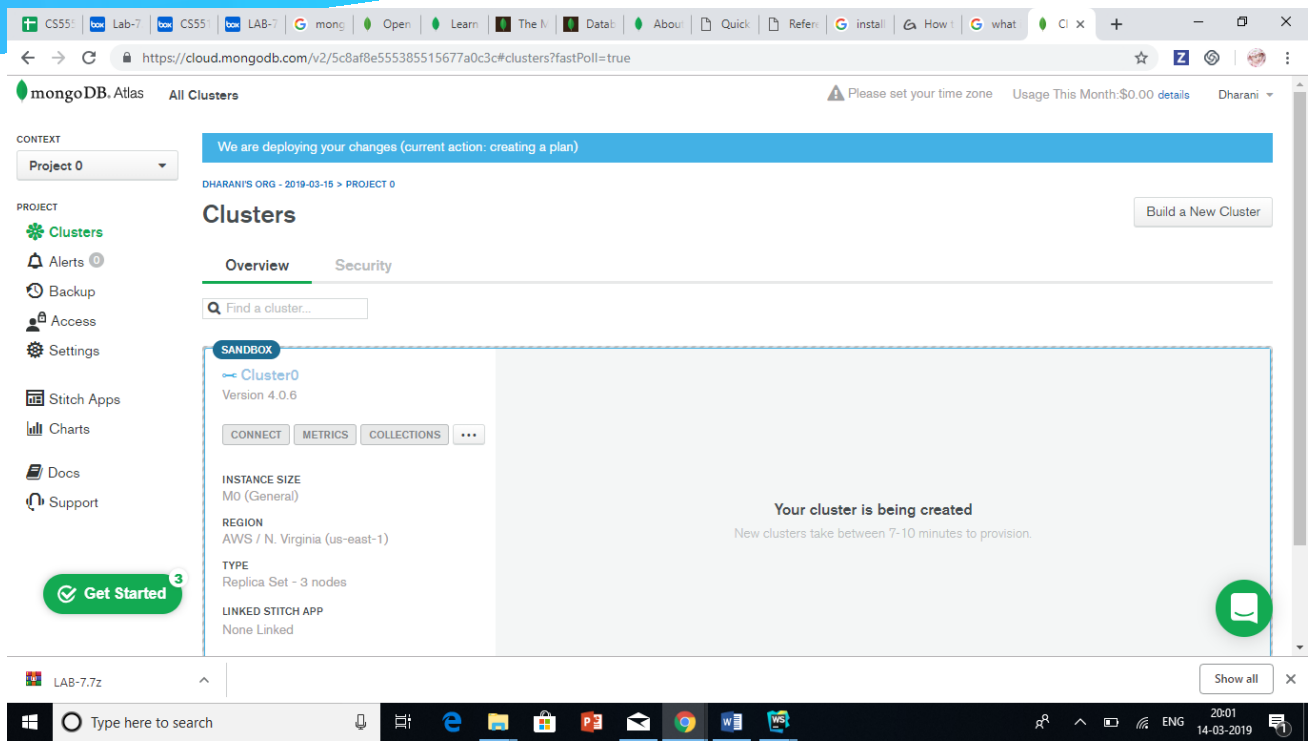
Step -1: We make sure all the pre-requisites are ready before start of the project and below are the technologies/languages used:

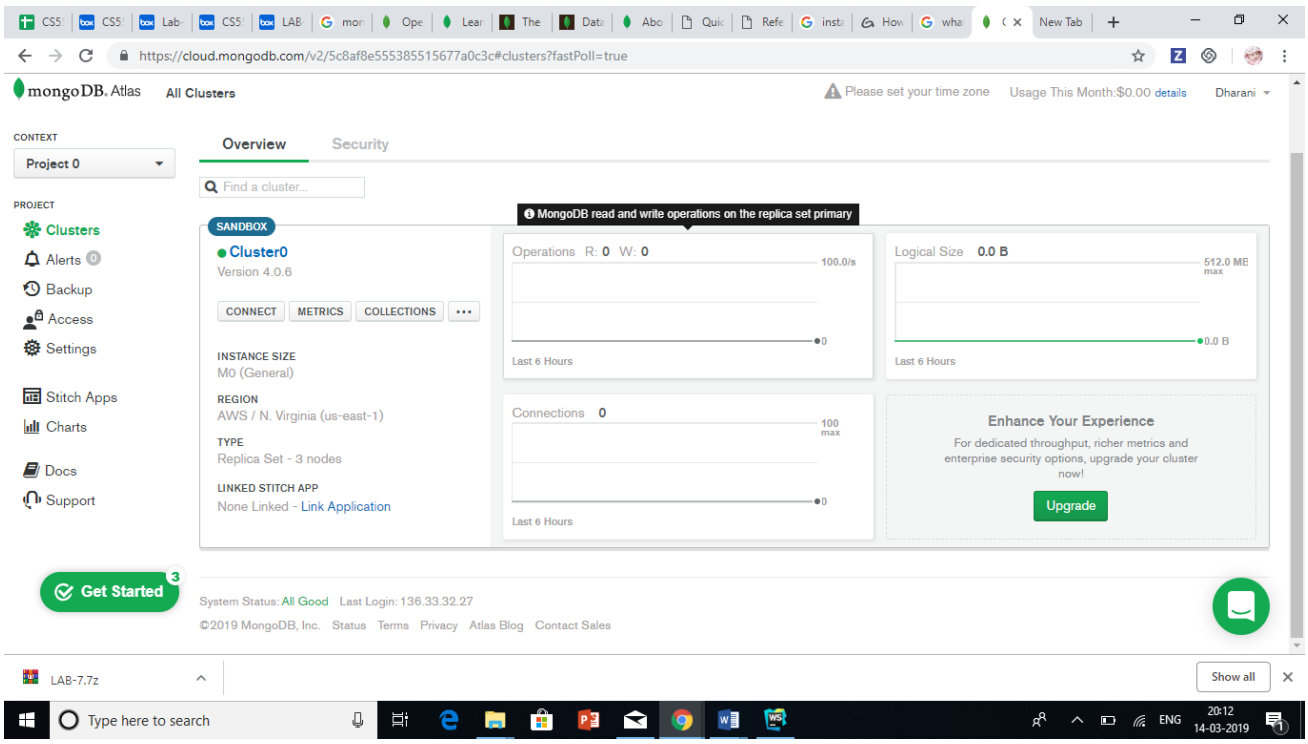
1. WebStorm IDE
2. Node
3. Npm
4. Expressjs
5. Angular
6. MongoDB: We should create Mongo Account and Create cluster. For that follow below Process:
    a. Create Account at https://cloud.mongodb.com



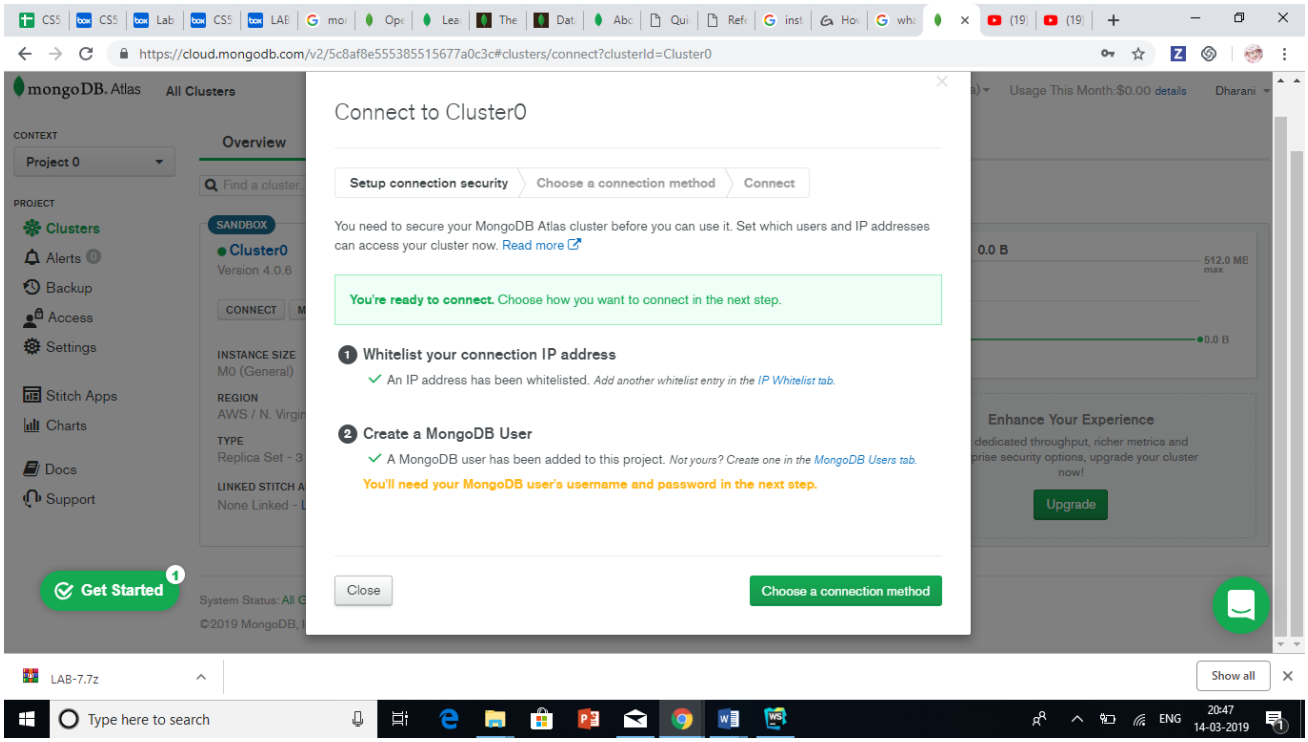

https://www.mongodb.com/atlas-signup-from-mlab
mongoDB Atlas
Almost Done
Please accept the MongoDB Atlas Terms of Service to complete your signup
I agree to the terms of service.
Create a MongoDB Atlas Account
Included with your free cloud database:
512 MB of Storage
Shared RAM
Highly available replica sets, end-to-end encryption, automated patches, REST API
Additionally, get access to the following when you launch a dedicated cluster:
10 GB or more of storage
Dedicated RAM
Performance optimization tools
Backups & point-in-time recovery
Enterprise security features including encryption key management, LDAP integration, and granular database auditing
Global Clusters
Learn more about MongoDB Atlas
LAB-7.7z
Show all
Type here to search
ENG
19:59
14-03-2019


https://cloud.mongodb.com/v2/5c8af8e555385515677a0c3c#clusters/edit
Cloud Provider & Region
AWS, N. Virginia (us-east-1)
aws
Google Cloud Platform
Azure
Create a free tier cluster by selecting a region with FREE TIER AVAILABLE and choosing the M0 cluster tier below.
★ recommended region
NORTH AMERICA
N. Virginia (us-east-1) ★
FREE TIER AVAILABLE
Ohio (us-east-2) ★
N. California (us-west-1)
Oregon (us-west-2) ★
FREE TIER AVAILABLE
Montreal (ca-central-1)
EUROPE
Stockholm (eu-north-1) ★
Ireland (eu-west-1) ★
FREE TIER AVAILABLE
London (eu-west-2) ★
Paris (eu-west-3) ★
Frankfurt (eu-central-1) ★
AUSTRALIA
Sydney (ap-southeast-2) ★
ASIA
Tokyo (ap-northeast-1) ★
Seoul (ap-northeast-2)
Singapore (ap-southeast-1) ★
FREE TIER AVAILABLE
$0.08/hour
Pay-as-you-go! You will be billed hourly and can terminate your cluster anytime. Excludes variable data transfer, backup, and taxes.
Cancel
Create Cluster
LAB-7.7z
Show all
Type here to search
ENG
20:01
14-03-2019


https://cloud.mongodb.com/v2/5c8af8e555385515677a0c3c#clusters?fastPoll=true
mongoDB. Atlas
All Clusters
Please set your time zone
Usage This Month:$0.00 details
Dharani
CONTEXT
Project 0
PROJECT
Clusters
Alerts 0
Backup
Access
Settings
Stitch Apps
Charts
Docs
Support
Get Started 3
We are deploying your changes (current action: creating a plan)
DHARANI'S ORG - 2019-03-15 > PROJECT 0
Clusters
Build a New Cluster
Overview
Security
Find a cluster...
SANDBOX
Cluster0
Version 4.0.6
CONNECT
METRICS
COLLECTIONS
INSTANCE SIZE
M0 (General)
REGION
AWS / N. Virginia (us-east-1)
TYPE
Replica Set - 3 nodes
LINKED STITCH APP
None Linked
Your cluster is being created
New clusters take between 7-10 minutes to provision.
LAB-7.7z
Show all
Type here to search
ENG
20:01
14-03-2019

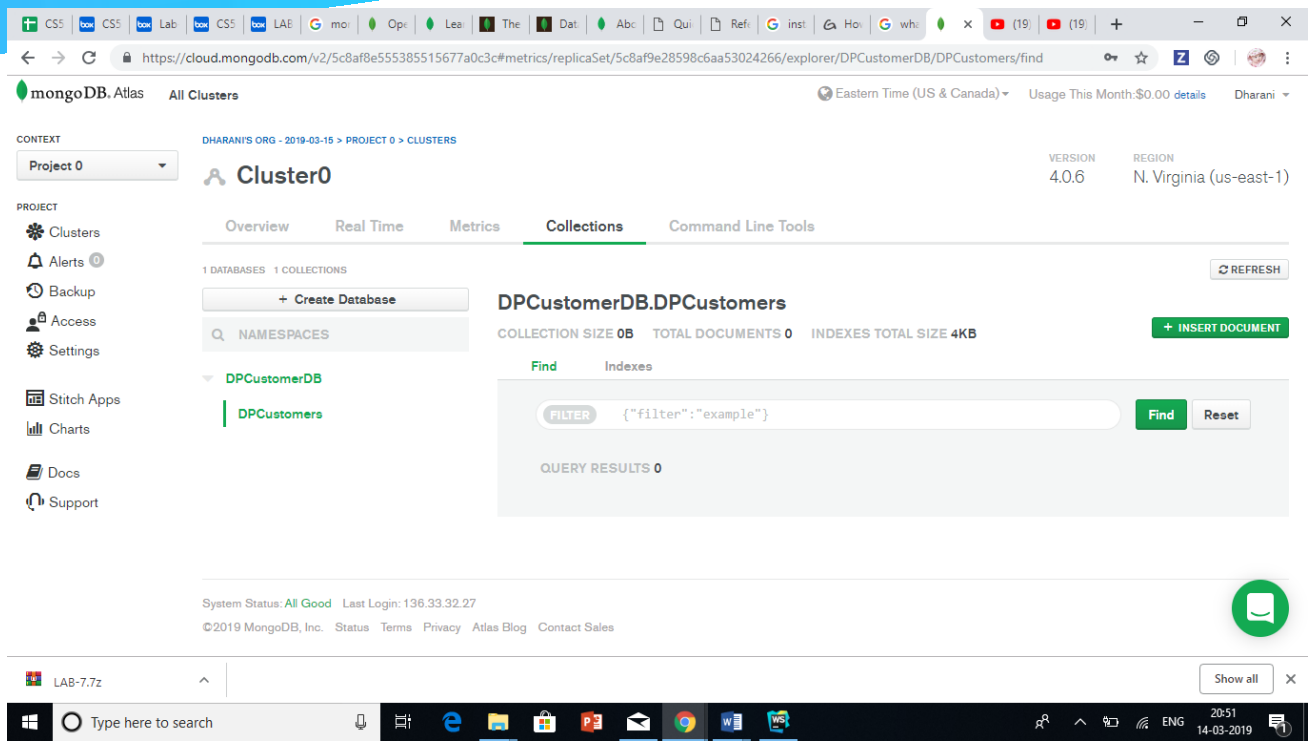

https://cloud.mongodb.com/v2/5c8af8e555385515677a0c3c#clusters?fastPoll=true
mongoDB. Atlas
All Clusters
Please set your time zone
Usage This Month:$0.00 details
Dharani
CONTEXT
Project 0
PROJECT
Clusters
Alerts 0
Backup
Access
Settings
Stitch Apps
Charts
Docs
Support
Get Started 3
Overview
Security
Find a cluster...
SANDBOX
Cluster0
Version 4.0.6
CONNECT
METRICS
COLLECTIONS
INSTANCE SIZE
M0 (General)
REGION
AWS / N. Virginia (us-east-1)
TYPE
Replica Set - 3 nodes
LINKED STITCH APP
None Linked - Link Application
MongoDB read and write operations on the replica set primary
Operations R: 0 W: 0
100.0/s
0
Last 6 Hours
Logical Size 0.0 B
512.0 MB max
0.0 B
Last 6 Hours
Connections 0
100 max
0
Last 6 Hours
Enhance Your Experience
For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!
Upgrade
System Status: All Good Last Login: 136.33.32.27
©2019 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales
LAB-7.7z
Show all
Type here to search
ENG
20:12
14-03-2019

b. We need to set up connection security by Create User Name and Password (remember those credentials as they will be useful in future). Moreover, select “Allow from anywhere” which will allow us to connect from anywhere by providing username and password. Then you setup is done



c. Choose connection method “Connect your Application” as we are planning to build an application and then close the window. In this page you can see connection string which will be useful to code and connect to MongoDB.

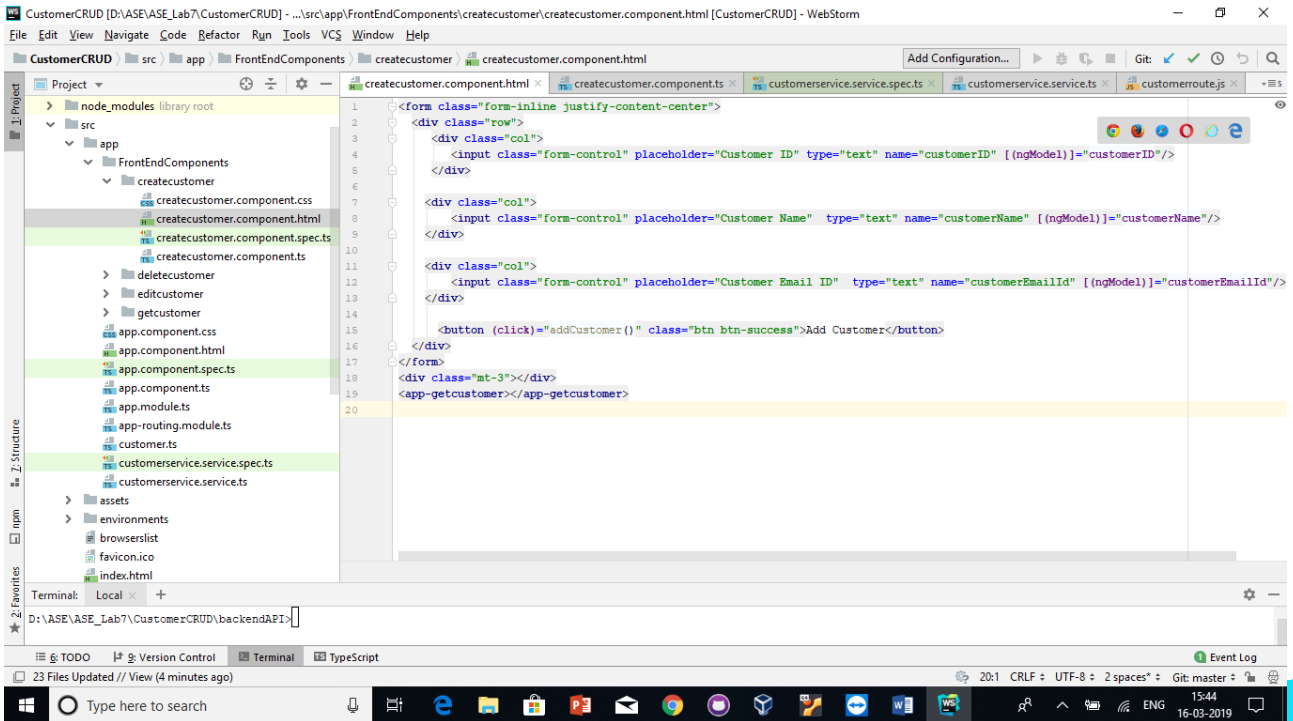
d. Click on “Collections” and Provide DB name and name of collection.

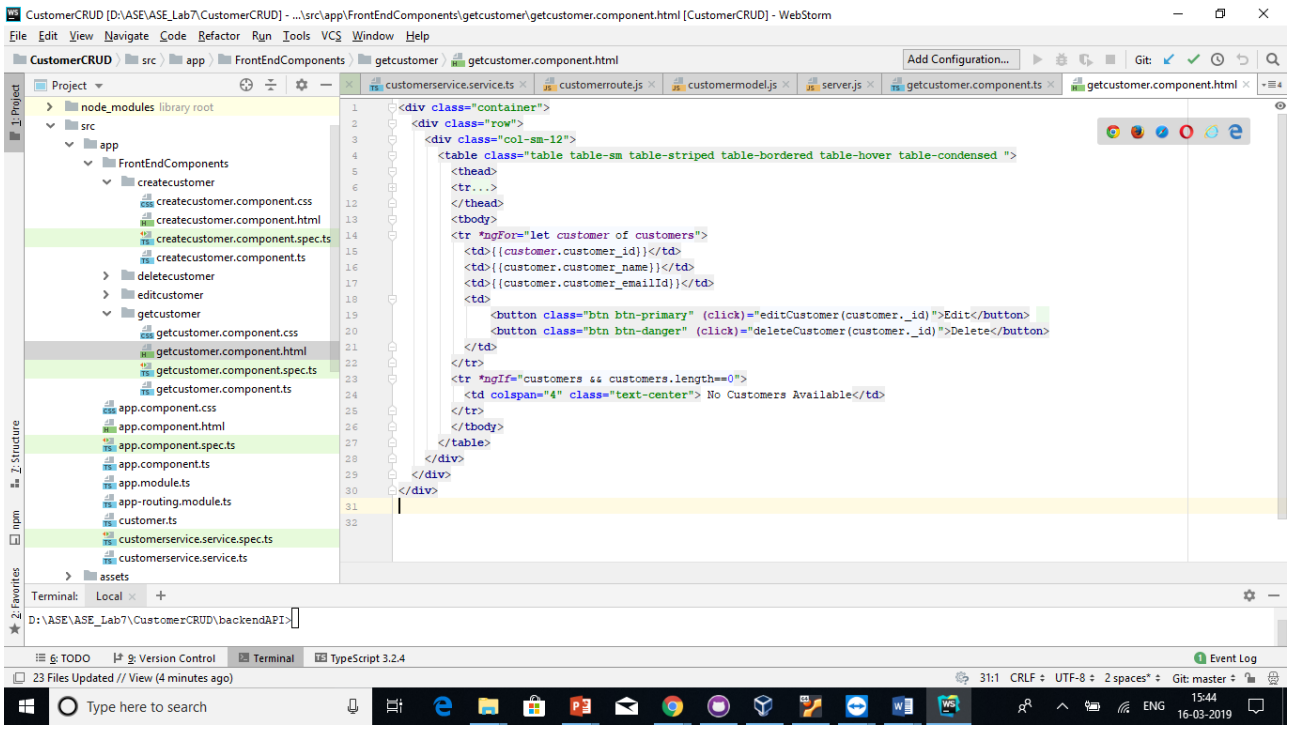


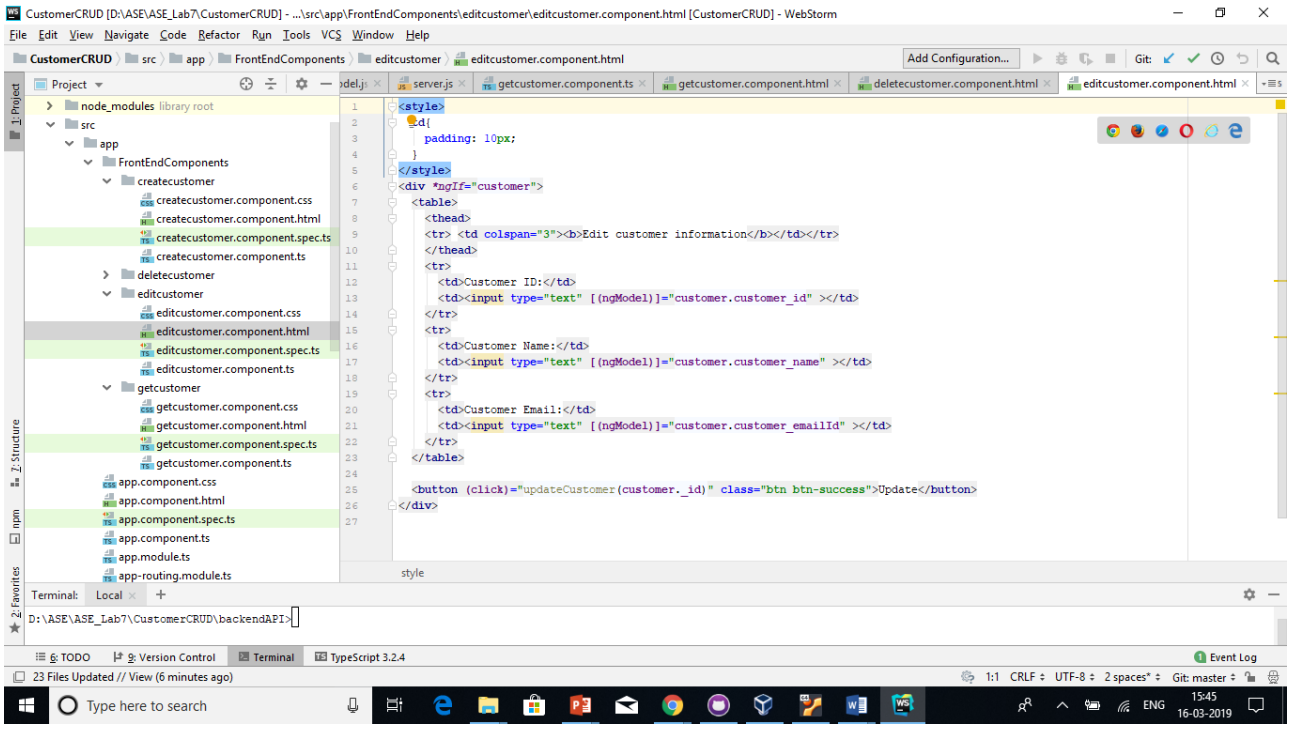
Step-2: As we are using Angular, we should create project using command " ng new DPCustomerProject" and Using this command I have created components :
ng g c components/customerlist

Step-3: We have done the required configurations for components in the app.routing.module.ts file

Step-4: We have written below html code for Customer list, Adding Customer, Editing and Deleting Customer:

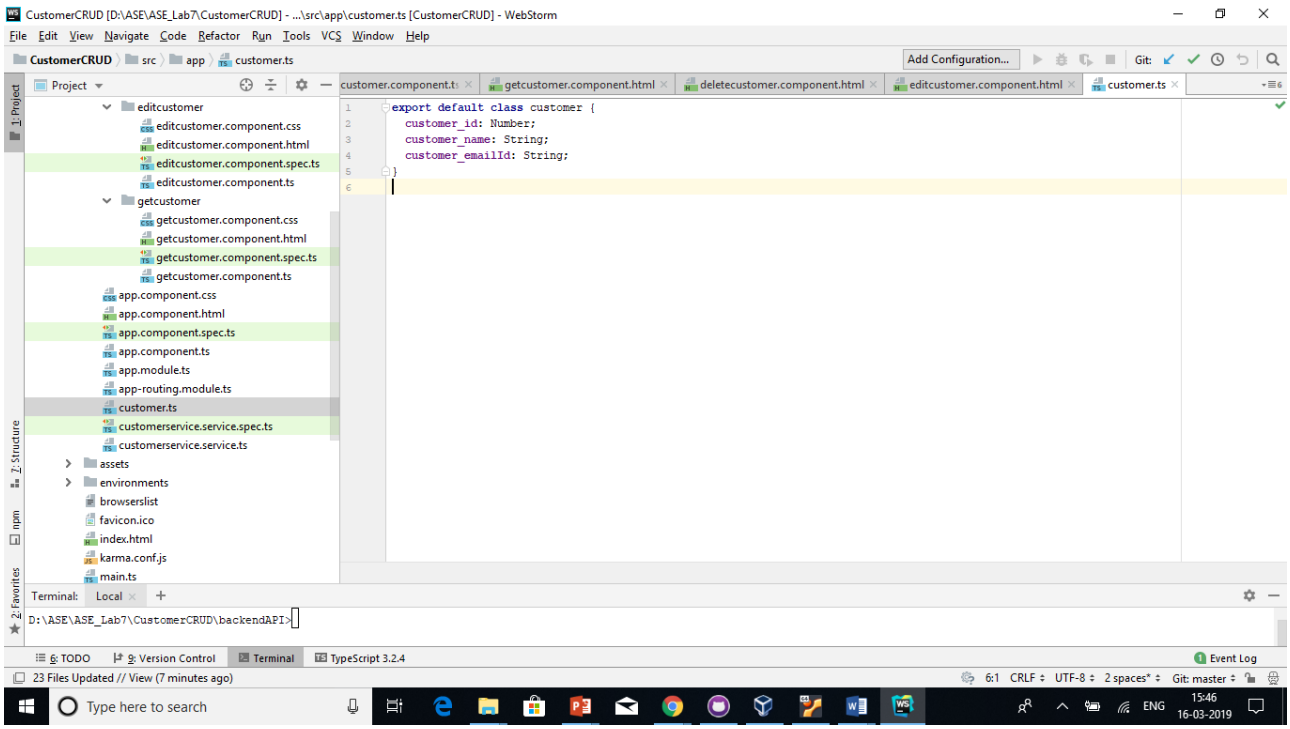




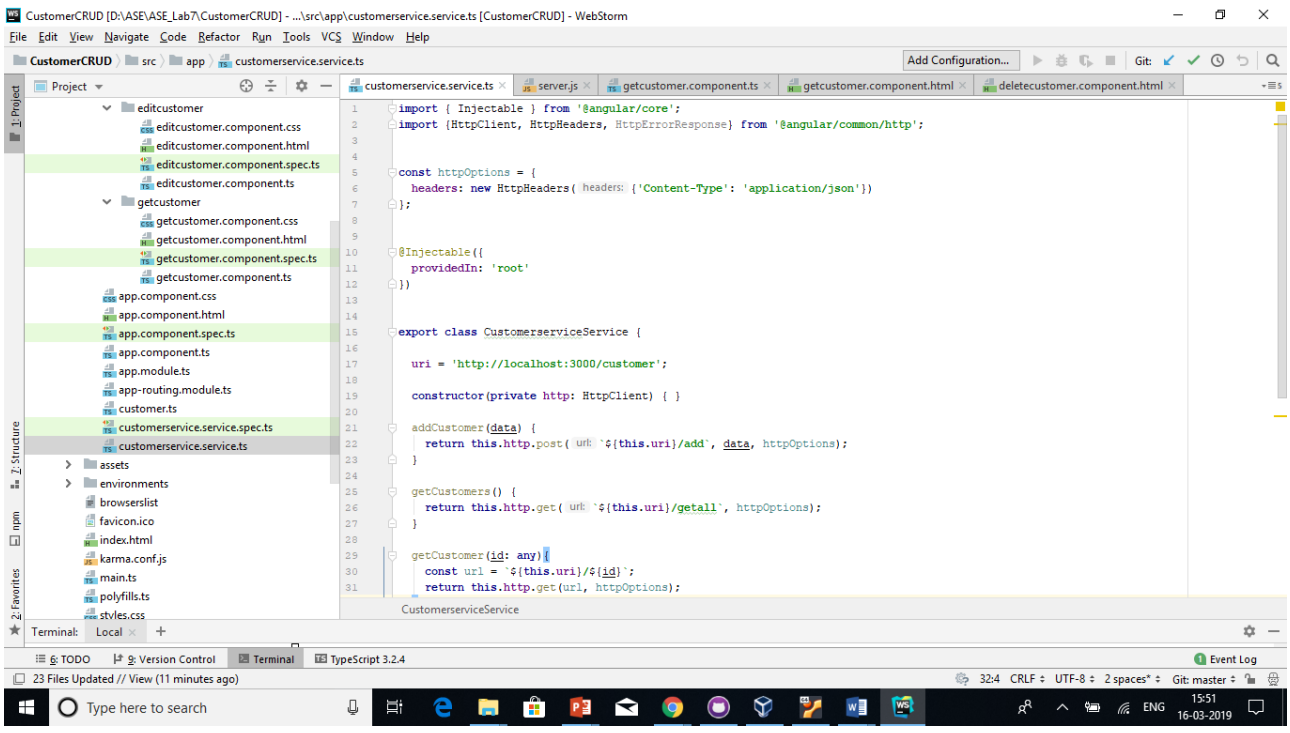


Step-5: Now we did the configuration of HTMLClientModule by importing the HTTPClientModule inside the app.module.ts file.

Step-6: Inside the src>>app folder we have created a file called customer.ts and add the following code to create a model at the front end

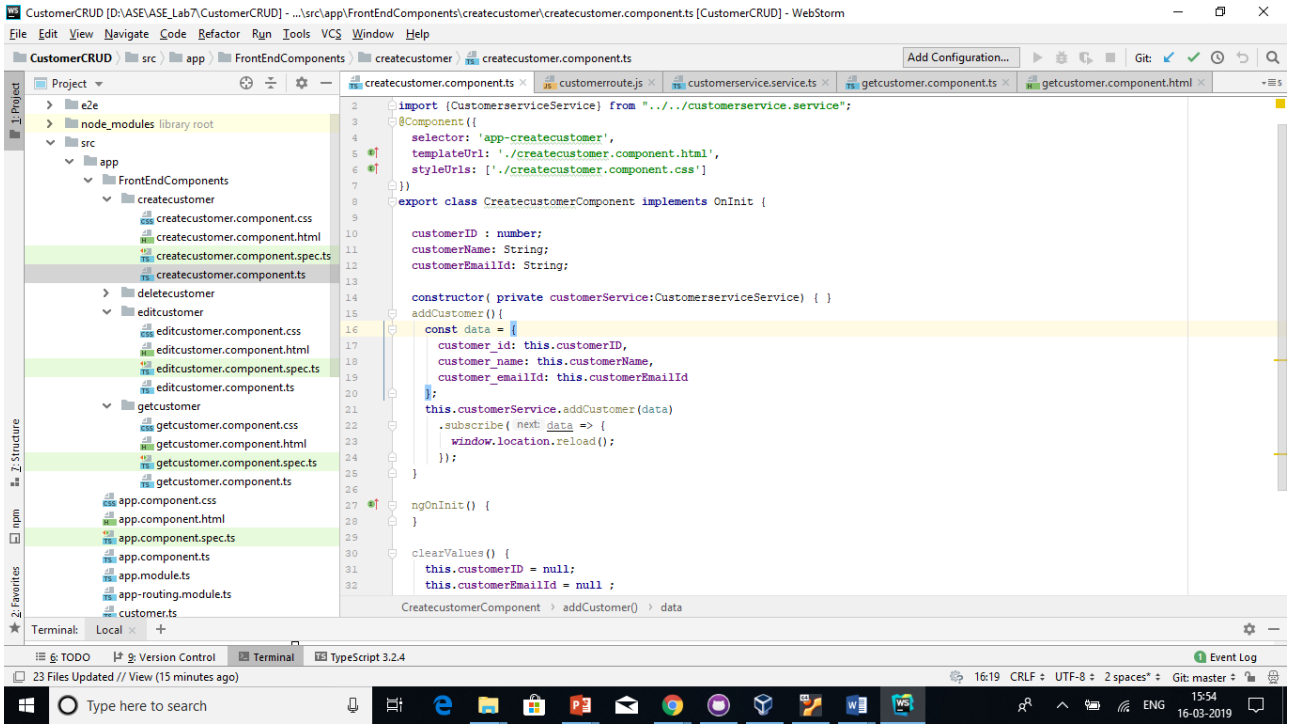


Step-7: Create angular service file by using the command “ng g service customerservice” and then imported this service file into app.module.ts. In this service we have written code that will send HTTP post request with the data to the Node.js server and save the data into MongoDB database.



We have defined out backend API URI but we haven’t created any backend yet in the initial stage but later we have done that.

Step-8: Create addCustomer function inside the createcustomer.component.ts file

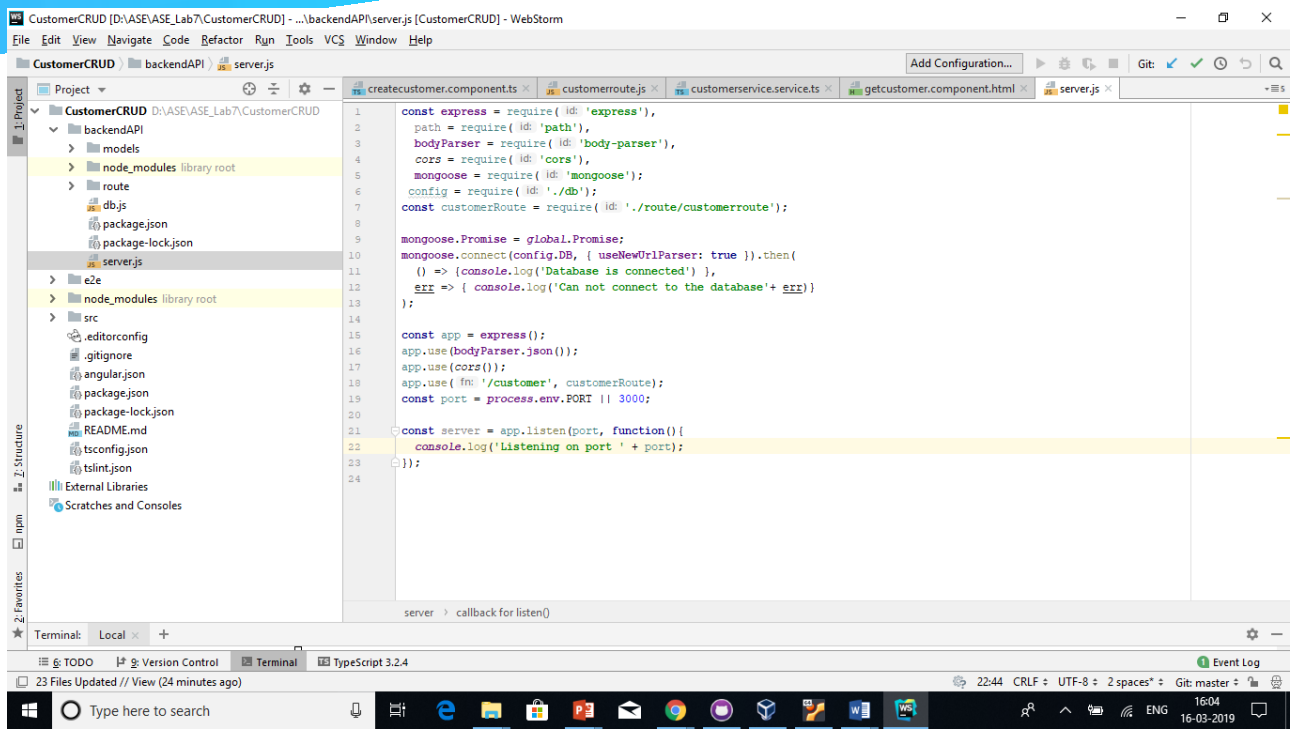


Here, we have defined the function and also imported
the **customerservice.service.ts** file. Instantiate the object inside the.

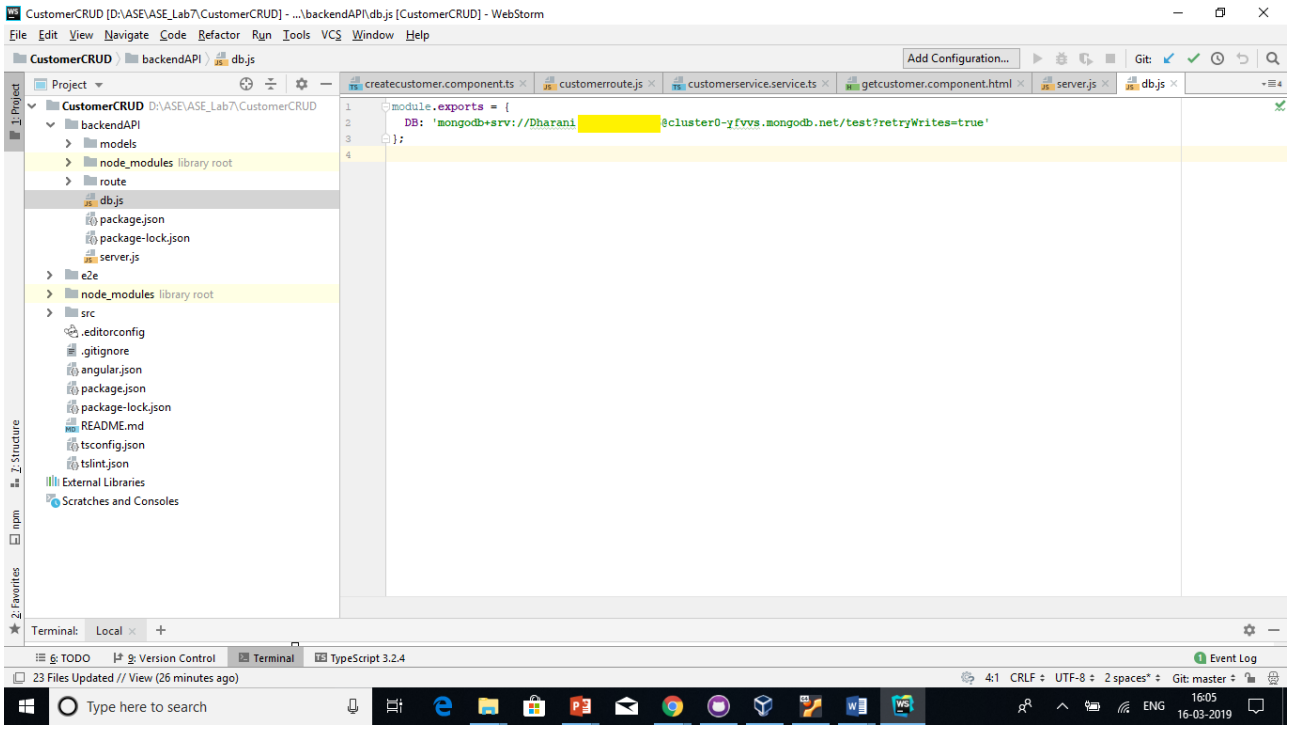
Step-9: Inside the angular root folder, create **backendAPI** folder and go inside that folder. Remember, it will be a completely separate project from Angular. So its **node_modules** are different from an **Angular**.

Open the terminal inside the **backendAPI** folder and type the following command : npm init
which will create package.json file and also installed the nodemodeules by using the command: npm install.

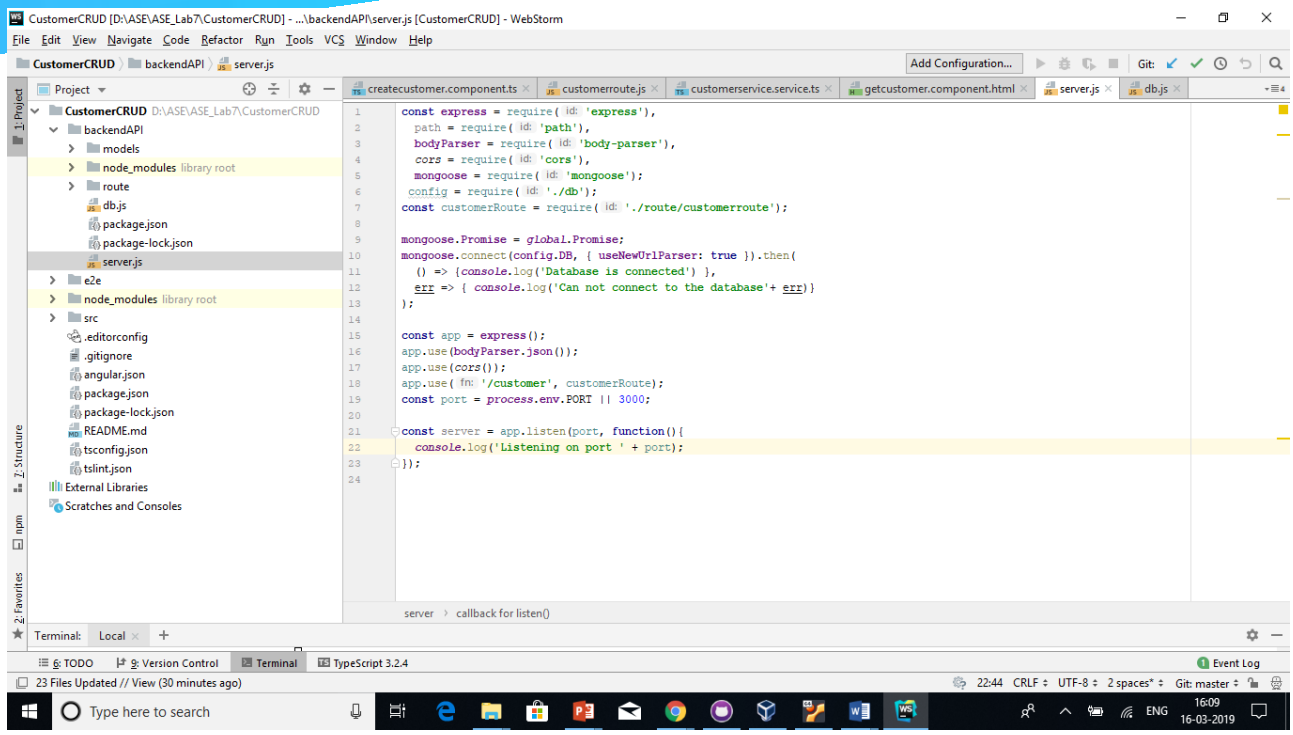
Step-10: Now, inside the **api** folder, create one file called the **server.js** file where in it contains the code to connect with port number 3000



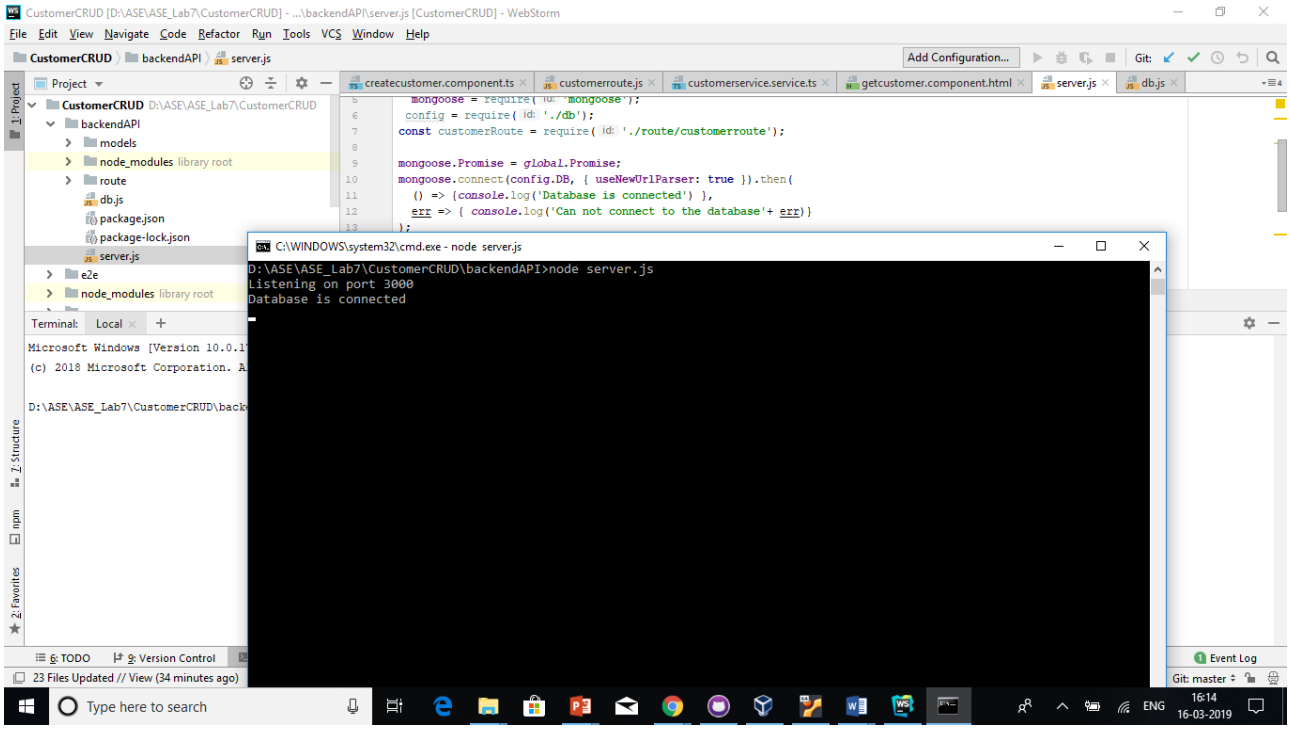
Step-11: Next thing is to connect MongoDB database with our node.js application.To achieve that we have created one file called **db.js** inside **backendAPI** root project folder. Written the following code inside the **db.js** file.



Step-12: Written the following code inside the **server.js** file to connect our **MongoDB** application to the **Node.js** server.



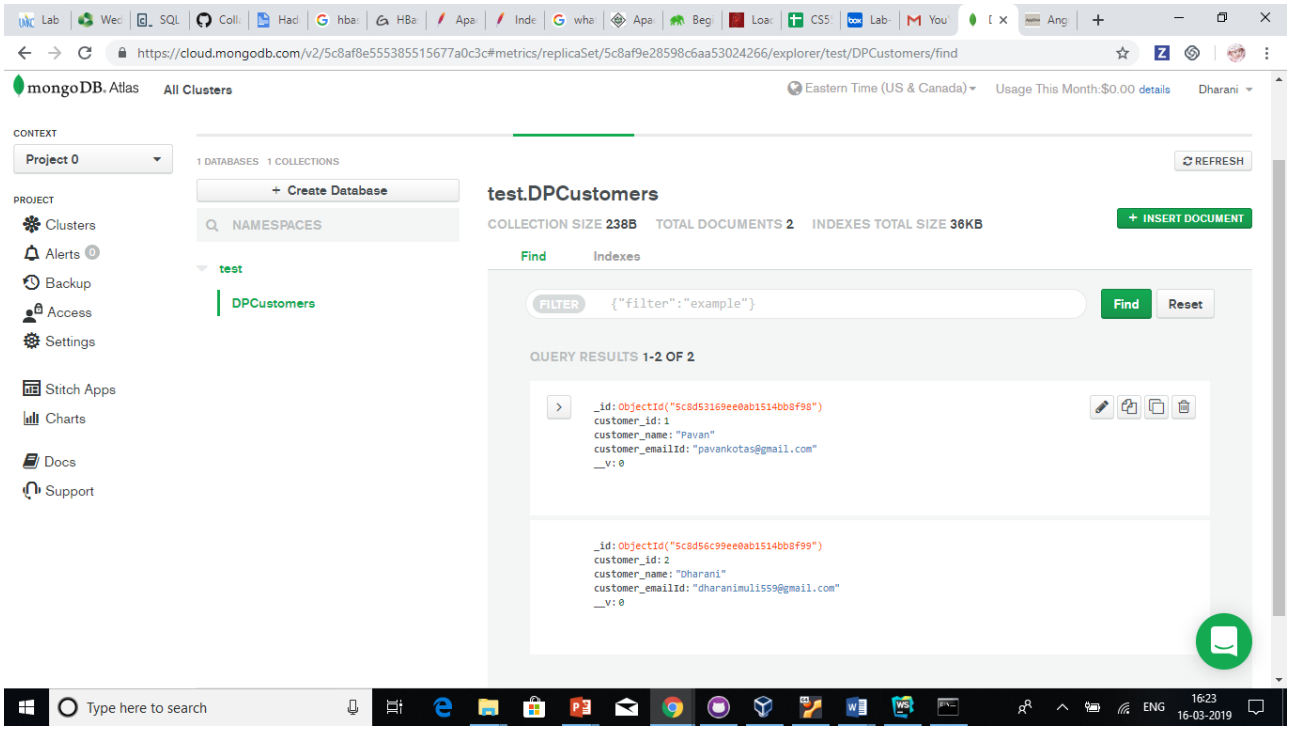
Step-13: Save our files and go to the terminal and start the node server to check weather we are able to connect to node and database or not



Step-14: Now created 2 folders inside the backendAPI root folder called route and models. In the models we have created custoemrmodels.js file and in the route customerroute.js file and we have defined our schema for the customer collection. We have three fields called **customer_ID, customer_Name and customer_EmailID.**

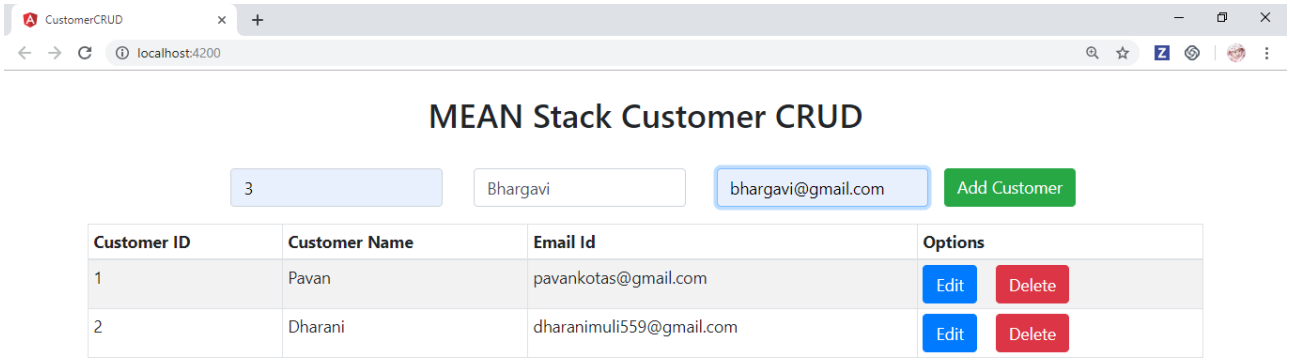
And created the CRUD code inside the **business.route.js** file.

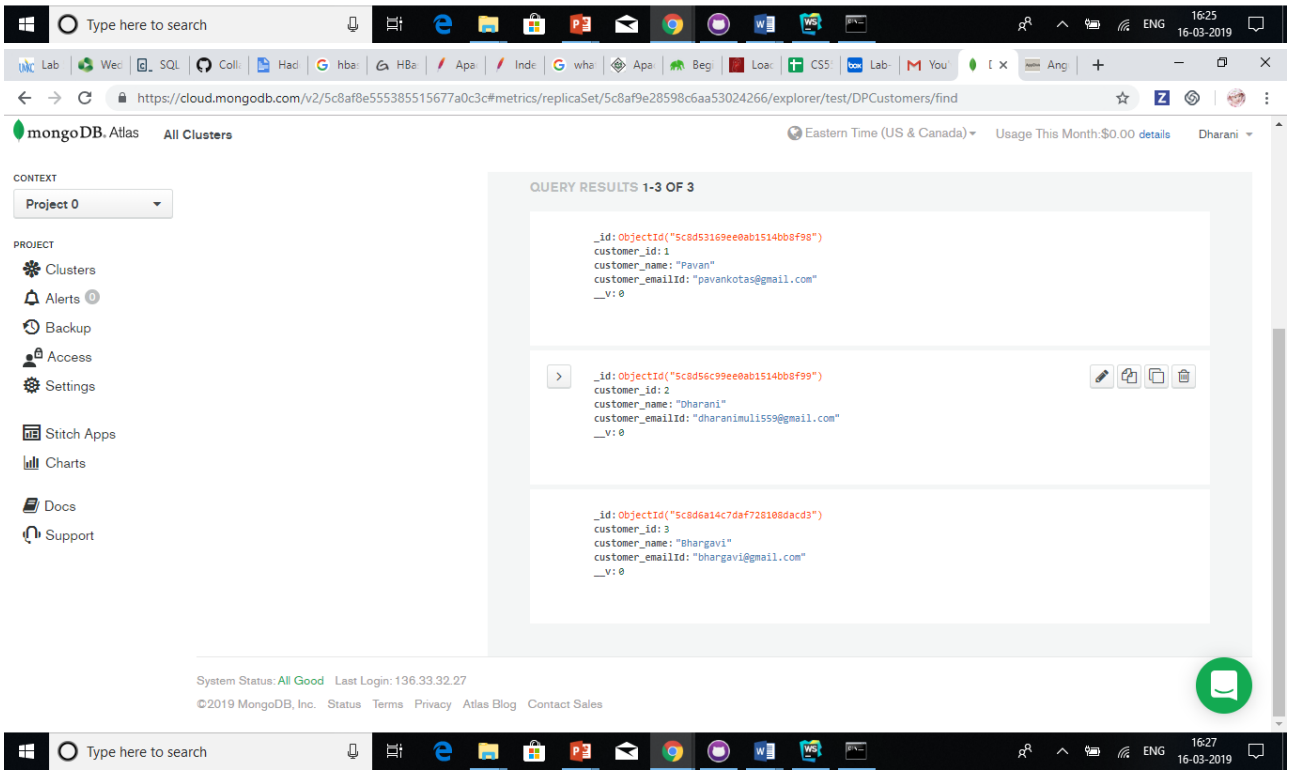
Step-15: We have created code under components.html files to display the data on the frontend and after running our code using command “ng serve -o” we can view our UI page where in when user add data in the input fields user can view data is getting into table, that says it is also getting added in MongoDB successfully.



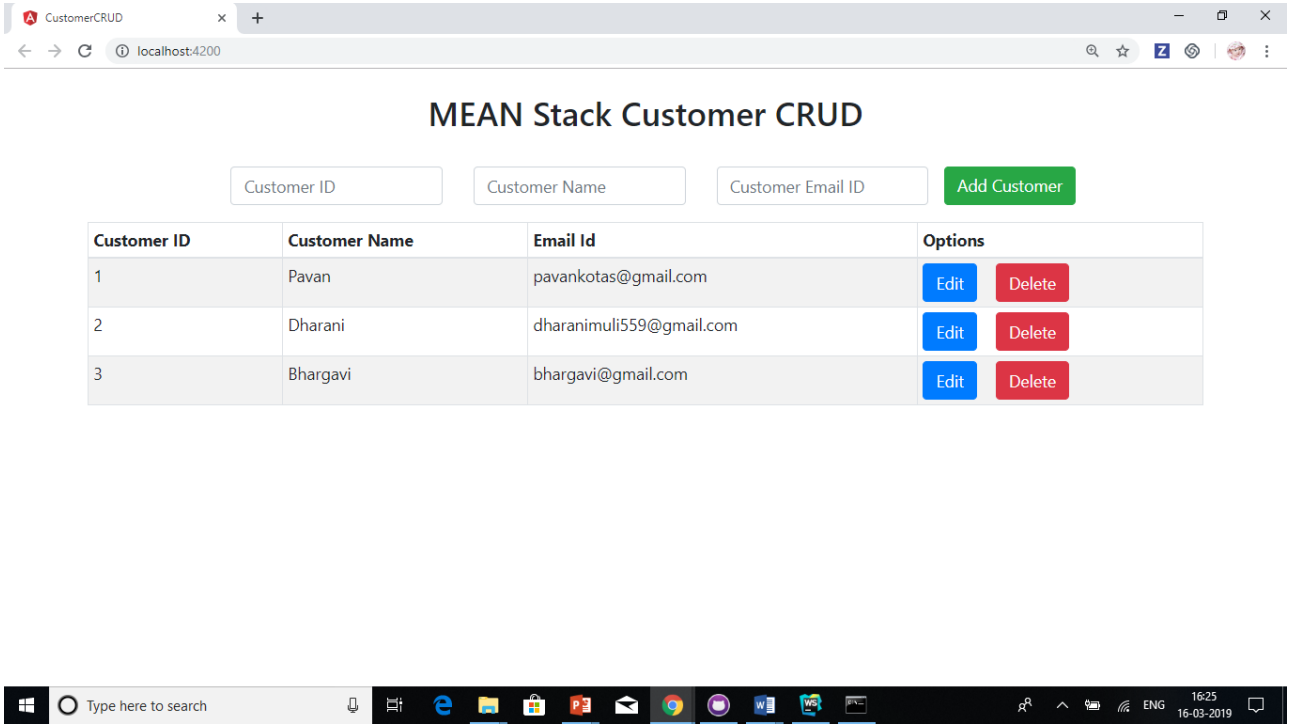
Sample output

1. Adding Customer

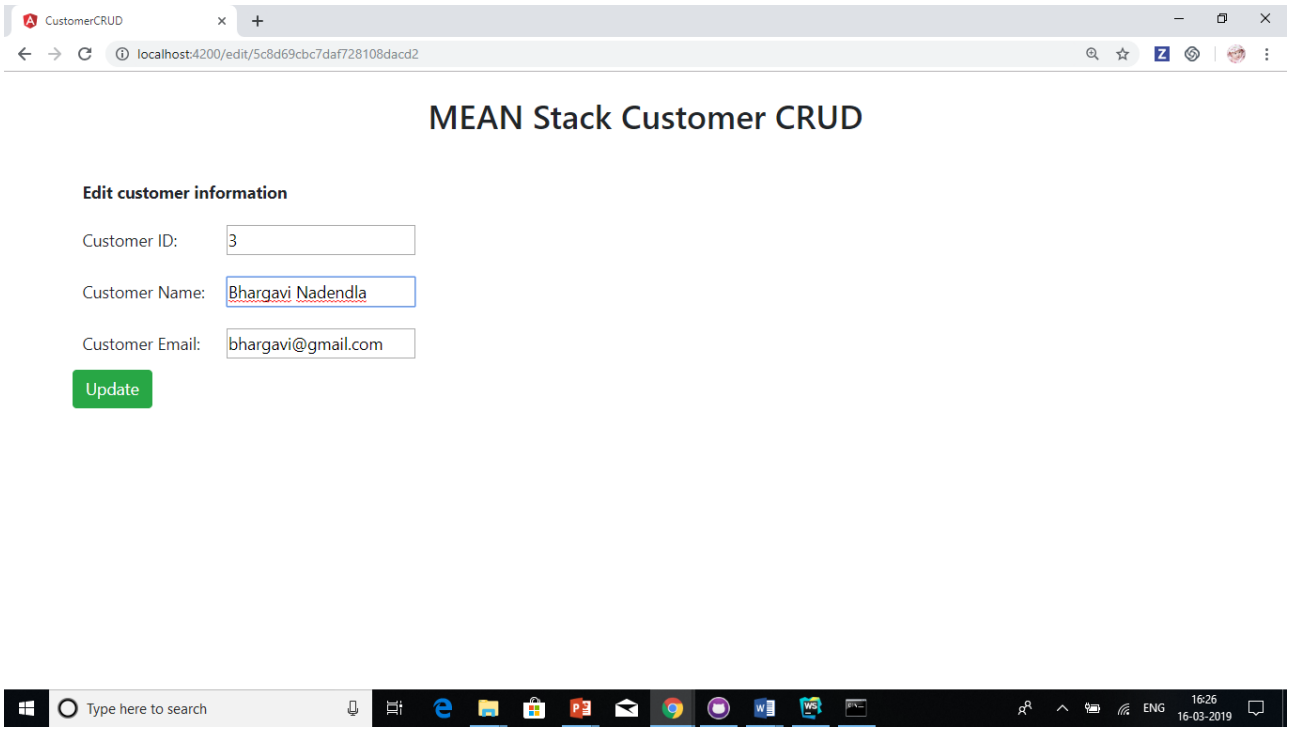


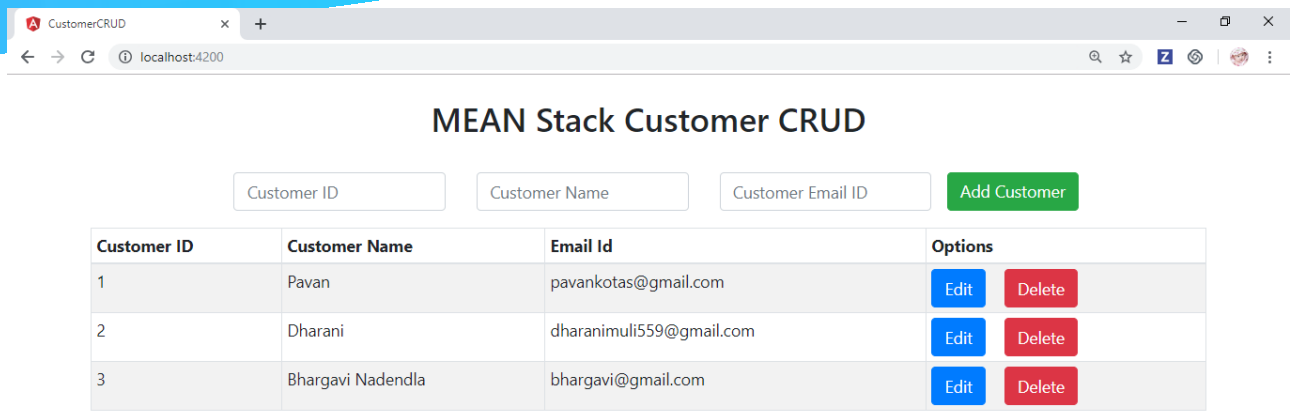


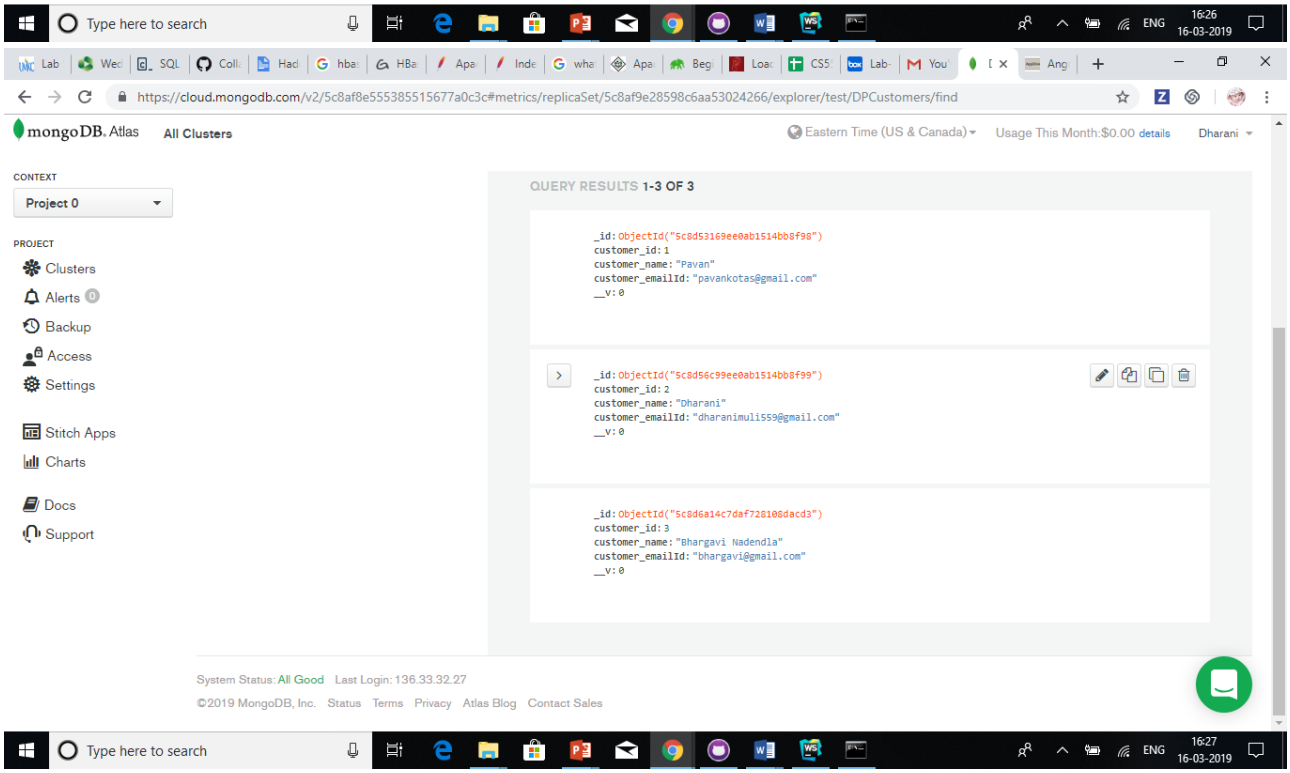
2. View the Customer in the table



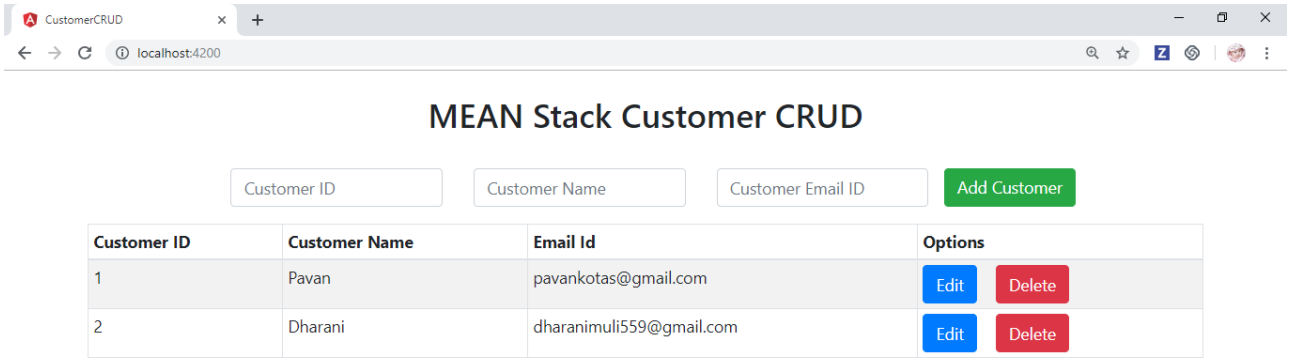
3. Editing the customer

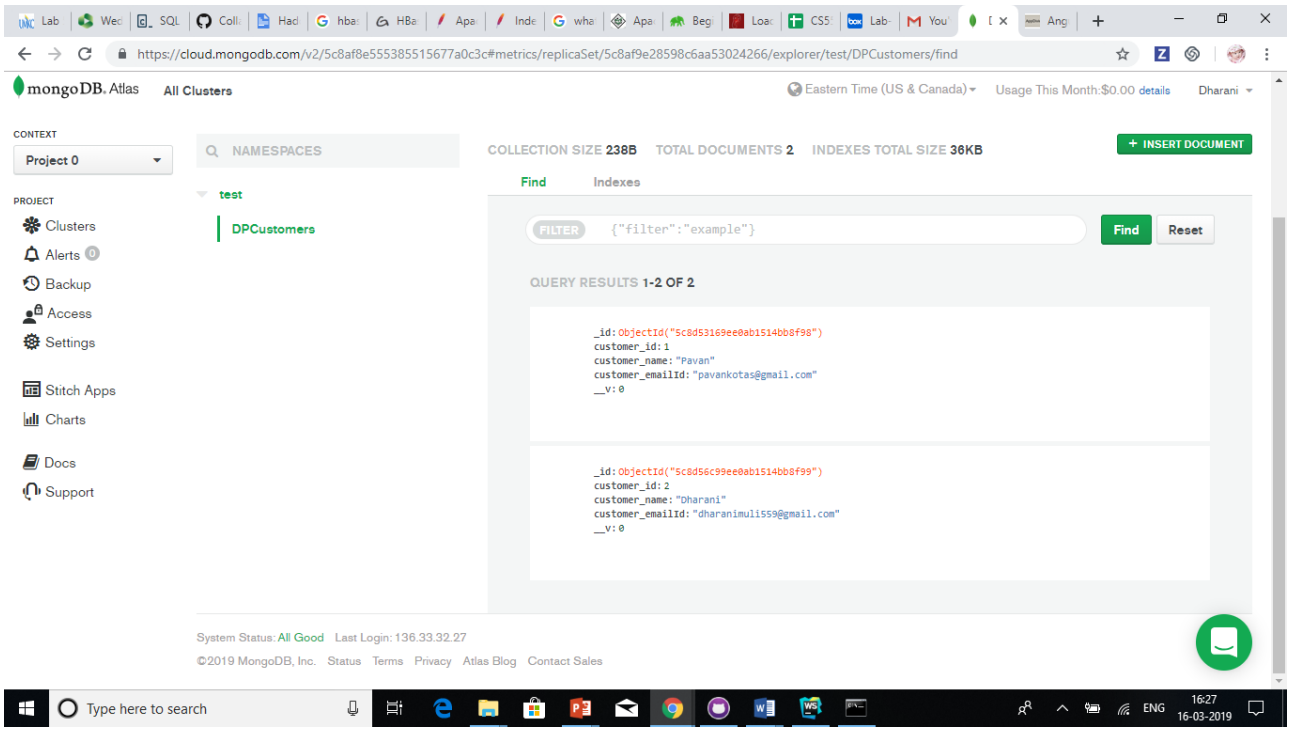



CustomerCRUD
localhost:4200
MEAN Stack Customer CRUD
Customer ID
Customer Name
Customer Email ID
Add Customer
Customer ID
Customer Name
Email Id
Options
1
Pavan
pavankotas@gmail.com
Edit
Delete
2
Dharani
dharanimuli559@gmail.com
Edit
Delete
3
Bhargavi Nadendla
bhargavi@gmail.com
Edit
Delete


Type here to search
https://cloud.mongodb.com/v2/5c8af8e555385515677a0c3c#metrics/replicaSet/5c8af9e28598c6aa53024266/explorer/test/DPCustomers/find
mongoDB. Atlas
All Clusters
Eastern Time (US & Canada)
Usage This Month:$0.00 details
Dharani
CONTEXT
Project 0
PROJECT
Clusters
Alerts
Backup
Access
Settings
Stitch Apps
Charts
Docs
Support
QUERY RESULTS 1-3 OF 3
_id: ObjectId("5c8d53169ee0ab1514bb8f98")
customer_id: 1
customer_name: "Pavan"
customer_emailId: "pavankotas@gmail.com"
__v: 0
_id: ObjectId("5c8d56c99ee0ab1514bb8f99")
customer_id: 2
customer_name: "Dharani"
customer_emailId: "dharanimuli559@gmail.com"
__v: 0
_id: ObjectId("5c8d6a14c7daf728108dacd3")
customer_id: 3
customer_name: "Bhargavi Nadendla"
customer_emailId: "bhargavi@gmail.com"
__v: 0
System Status: All Good
Last Login: 136.33.32.27
©2019 MongoDB, Inc.
Status
Terms
Privacy
Atlas Blog
Contact Sales
Type here to search

4. Deletion of customer





Issues/Limitations:

1. Faced few problem while connecting and doing CRUD operations in Mondo DB but later we successfully resolved all the problems and able to complete the assignment.
2. We have not much focused on the designing UI because main aim of this project is to build CRUD operations using MEAN stack with the basic frontend. But we have done basic good looking UI not the best looking.

Team Contribution

Dharani:

1. Created MongoDB account and established its connection
2. Implemented “Add” and “Listing” functionalities and also created interactive commands for the same.
3. Contributed in the creation of Wiki page, report and lab submission.

Chakra Pavan Kumar Kota:

1. Helped in resolving issues while connecting to MongoDB and saving data.
2. Implemented “Update” and “Remove” functionalities.
3. Contributed creation of Wiki page, report and lab submission.

Conclusion:

With this exercise we are able to understand the way CRUD operations can be performed in Node.js. We are able to analyse the previous source code provided and reapply the skills while implementing the CRUD repository for Customers. This javascript based node.js approach to access files, update them seemed to be very exciting which is otherwise impossible in plain JavaScript world. This also helps us in incorporating the skills in the final project whose backend is Node.js.